e-Informatica

software engineering journal

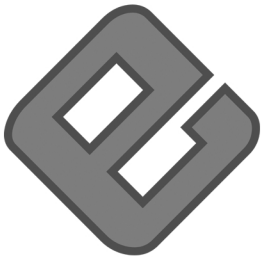
2018 volume 12 issue 1

e-Informatica

e-Informatica

software engineering journal

2018 volume 12 issue 1



e-Informatica

Wrocław University
of Science and Technology

Editors
Zbigniew Huzar (*Zbigniew.Huzar@pwr.edu.pl*)
Lech Madeyski (*Lech.Madeyski@pwr.edu.pl, http://madeyski.e-informatyka.pl*)

Department of Software Engineering, Faculty of Computer Science and Management,
Wrocław University of Science and Technology, 50-370 Wrocław, Wybrzeże Wyspiańskiego 27,
Poland

e-Informatica Software Engineering Journal
www.e-informatyka.pl, DOI: 10.5277/e-informatica

Editorial Office Manager: Wojciech Thomas

Proofreader: Anna Tyszkiewicz

Typeset by Wojciech Myszka with the LaTeX 2$_\varepsilon$ Documentation Preparation System

OFICYNA WYDAWNICZA POLITECHNIKI WROCŁAWSKIEJ
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław
www.oficyna.pwr.edu.pl;
e-mail: oficwyd@pwr.edu.pl; zamawianie.ksiazek@pwr.edu.pl

ISSN 1897-7979

Print and binding: beta-druk, www.betadruk.pl

Editorial Board

Gratitude for Reviewers

We would like to express appreciation to all reviewers for the effort and expertise contributed to reviewing, without which it would be difficult to maintain and raise the high standard of our peer-reviewed journal.

Tore Dybå
Sousuke Amasaki
Saïd Assar
Pearl Brereton
Steve Counsell
Darko Durisic
Robert Feldt
Vincenzo Ferme
Dariusz Gall
Jarosław Hryszko
Sami Hyrynsalmi
Foutse Khomh
Barbara Kitchenham
Sylwia Kopczyńska
Mathieu Lavallée
Luigi Lavazza
Valentina Lenarduzzi
Lech Madeyski
Fuensanta Medina Domínguez
Nasir Minhas
Jefferson Molleri
Marta Olszewska
Mel Ó Cinnéide
Fabio Palomba
Kai Petersen
Marcin Pietranik
Pierre Robillard
Per Runeson
Aneesa Saeed
Faiz Shah
Miroslaw Staron
Davide Taibi
Guilherme Travassos
Adam Trendowicz
Masateru Tsunoda
Magdalena Turowska
Michael Unterkalmsteiner
Sira Vegas
Auri Vincenzi
Anita Walkowiak
Bartosz Walter
Dietmar Winkler
Franz Wotawa
Andrzej Zalewski

Contents

e-Informatica Software Engineering Journal, Volume 12, Issue 1, 2018, pages: 9–26, DOI 10.5277/e-Inf180101

A Graphical Modelling Editor for STARSoC Design Flow Tool Based on Model Driven Engineering Approach

Elhillali Kerkouche*, El Bay Bourennane**, Allaoua Chaoui***

*Department of Computer Science, Mohamed Seddik Ben Yahia University, Jijel, Algeria

**LE2I Laboratoire, University of Bourgogne, Dijon, France

***MISC Laboratory, Department of Computer Science and its Applications, Faculty of IT, Abdelhamid Mehri University, Constantine, Algeria

elhillalik@yahoo.fr, ebourenn@u-bourgogne.fr, a_chaoui2001@yahoo.com

Abstract

Background: Due to the increasing complexity of embedded systems, system designers use higher levels of abstraction in order to model and analyse system performances. STARSoC (Synthesis Tool for Adaptive and Reconfigurable System-on-Chip) is a tool for hardware/software co-design and the synthesis of System-on-Chip (SoC) starting from a high level model using the StreamsC textual language. The process behaviour is described in the C syntax language, whereas the architecture is defined with a small set of annotation directives. Therefore, these specifications bring together a large number of details which increase their complexity. However, graphical modelling is better suited for visualizing system architecture.

Objectives: In this paper, the authors propose a graphical modelling editor for STARSoC design tool which allows models to be constructed quickly and legibly. Its intent is to assist designers in building their models in terms of the UML Component-like Diagram, and in the automatic translation of the drawn model into StreamsC specification.

Methods: To achieve this goal, the Model-Driven Engineering (MDE) approach and well-known frameworks and tools on the Eclipse platform were employed.

Conclusion: Our results indicate that the use of the Model-Driven Engineering (MDE) approach reduces the complexity of embedded system design, and it is sufficiently flexible to incorporate new design needs.

Keywords: embedded systems, hardware/software co-design, STARSoC tool, UML, model-driven engineering, Eclipse modelling project

1. Introduction

The increasing complexity of embedded system designs calls for high level specification languages (like StreamsC [1] or others C/C++ based extensions), and for automated transformations towards lower level descriptions. These languages allow to create high level models quickly, run simulations, optimize designs and investigate the efficiency of different algorithms and architectures before generating their corresponding low level implementations. The automatic generation of low level implementation drastically reduces the amount of code to be written by designers, which saves time to market and reduces fabrication costs compared to hand-tuned implementations [2]. For these reasons, the design tools are widely adopted by the embedded system designers' community [3]. The specification of the applications becomes easier at high abstraction levels, since the implementation details are hidden from the designer.

The Synthesis Tool for Adaptive and Reconfigurable System-On-Chip(STARSoC) [4] is one

of those design tools that allow hardware-software co-design, design space exploration and high level synthesis from a StreamsC textual specification. The StreamsC language [5] permits the modelling of the architecture and the behaviour of a complex embedded system containing both Hardware and Software communicating processes. In StreamsC textual models, the architecture of the system is defined with a collection of annotation directives which are used to declare processes and communication between them, whereas processes' behaviours are described in the C programming language. Therefore, these specifications allow for gathering a lot of details (system architecture and processes' behaviours) which increase their length and their complexity, and consequently decrease their legibility.

It is well known that graphical specification is better suited for describing the system components and their relationships, whereas components' behaviours are generally expressed in textual notations (like the C programming language) which allow their reuse as building blocks in new designs. The optimal modelling solution consists in combining textual notations with graphical notations in order to accumulate their advantages. Thereby, every system aspect is provided with the most suitable view (textual/graphical). UML Component Diagrams [6] are widely used to define the structure of a system. A Component Diagram provides a clear view of the organization and the dependency among components in a system, including their contents (source code, binary code or executable) and their interfaces through which they interact with one another. In this work, the Authors propose to develop a graphical modelling editor for the STARSoC design tool. More precisely, it is an approach and a tool support to allow a high-level graphical specification of embedded systems which combines the architectural and behavioural aspects of a system in one model. The architectural aspect is expressed with a UML Component-like Diagram which is an adaptation of the UML Component Diagram to the structural concepts of the StreamsC language, whereas the behavioural aspect is specified in the C programming language. From the graphical specification of a system, this approach permits to automatically generate a clean and correct SteamsC specification. In order to achieve this objective, it is proposed to use the Model-Driven Engineering (MDE) [7] approach which is based on meta-modelling and Model Transformations, and to employ well-known frameworks and tools under the Eclipse platform to in this automatic approach.

The rest of the paper is organized as follows. Section 2 outlines the major related works. In Section 3, some concepts of the StreamsC language are presented. Section 4 presents the STARSoC Tool. In Section 5, an overview of the Eclipse Modelling Project is given. In Section 6, the approach is presented and it is applied on an example in Section 7. The last section concludes the paper and gives some perspectives of this work.

2. Related works

In the literature, several research works have been done on the automatic code generation tools for Multi-Processor Systems-on-Chip (MPSoCs) in order to facilitate and to accelerate the design process.

In addition to STARSoC, there are several code generation tools for MPSoCs which use the textual specification of the whole system as input. From this high level specification containing various system parameters, the tools generate a low level description of the system and perform their functionalities which are necessary in the design process, such as simulation, design space exploration, performance evaluation, etc. For example, xENOC [8] is an automatic environment for hardware/software design of Network-on-Chip (NoC)-based MPSoC architectures. xENoC is based on a tool, called NoCWizard which uses an eXtensible Markup Language (XML) specification (including NoC features, Intellectual Properties (IPs) and mapping) to generate many types of NoC instances by using Verilog language [9]. In addition to NoC instances generation, xNoC also includes an Embedded Message Passing Interface (eMPI) supporting parallel task communication. SystemCoDesigner [10] is another design environment for high-level system modelling and simulation, automatic design space exploration

and automatic hardware/software synthesis from abstract model to final implementation. In SystemCoDesigner, the input model is given using SystemC textual language [11] which describes the structural and behavioural aspects of the system. In addition to academic environments, some commercial design environments support the creation of MPSoCs. The most popular are Altera System on a Programmable Chip (SoPC) [12] and Xilinx Embedded Development Kit (EDK) [13]. In these environments, the hardware part description and the hardware-software integration of the final system are strongly automated using an extensive IP cores library. Although textual notations better describe system parameters and aspects for the design and implementation, these notations increase the complexity of system specifications.

On the other hand, several research works have been proposed to adapt the UML notation to the modelling of embedded systems. The advantage of UML is that it can be extended to any particular domain by defining profiles which introduce additional domain-specific modelling concepts and constraints. In this context, many profiles have been proposed for embedded systems design. The SysML (System Modelling Language) profile [14] reuses a subset of UML notation and provides additional extensions needed in system engineering. It offers graphical modelling support for the specification, analysis, design, verification and validation of complex heterogeneous systems that may combine hardware and software components. The MARTE (Modelling and Analysis of Real-time and Embedded Systems) profile [15] is another UML profile which adds capabilities to UML for the development of Real Time and Embedded Systems (RTES). This extension provides support for specification, design and verification/validation phases. In addition, it defines a common way of modelling both the hardware and software aspects of systems (such as the representation of repetitive structures) in order to improve communication between developers. In order to cope with the design complexities of intensive signal and image data processing applications, the DaRT (Data-parallelism for Real-Time) team [16] of LIFL (the Computer Science Laboratory of Lille University, French) developed a design flow methodology and a tool labelled GASPARD2 [17]. Using a subset of MARTE Profile, GASPARD2 follows the Model Driven Architecture (MDA) [18] principles to describe systems at different level of abstractions. It emphasizes system level co-modelling (hardware and software), simulation, models refinement, automatic code generation and IPs integration. The UML-SystemC profile [19] is proposed to take advantages of both UML and the SystemC language. It captures both the structural and the behavioural features of the SystemC language and allows high level modelling of systems with straightforward translation to the SystemC code. In [20], the authors proposed an UML-based design environment, called Koski, for MPSoCs implementations of wireless sensor network applications. It provides a complete design flow covering the design phases from system level modelling to the FPGA (Field Programmable Gate Array) prototyping. Note that only the relevant profiles have been given here. Many other works which combine the UML modelling with embedded system design flow exist in the literature. However, they rarely cover all design phases from requirement modelling to implementation and validation.

In this work, the Authors intend to introduce a straightforward graphical modelling layer for the STARSoC tool. The proposed graphical modelling editor increases flexibility by integrating the UML notation (UML Component Diagram notation) to the STARSoC input specification language (StreamsC). Furthermore, it takes advantages of the MDE approach to rapidl design systems and integrates new design needs.

3. StreamsC language

The StreamsC language is a parallel programming language following the communicating process model [5]. The language is a small set of directives and library functions callable from a conventional C program. The directives are used to declare three distinguished objects: process, stream or signal, whereas the library functions

```
/// PROCESS_FUN <function_name>
/// IN_STREAM <stream element_data_type> <stream_name>
/// OUT_STREAM <stream element_data_type> <stream_name>
/// IN_SIGNAL <signal element_data_type> <signal_name>
/// OUT_SIGNAL <signal element_data_type> <signal_name>
/// PROCESS_FUN_BODY
. . . . . .
... C code ...
. . . . . .
/// PROCESS_FUN_END
```

Figure 1. Format of the PROCESS FUN directive

```
/// PROCESS <process_name> PROCESS_FUN <process_fun_name> [TYPE [SP | HP]] <on_spec>
```

Figure 2. Format of a process directive

are used to communicate stream data between processes. In the StreamsC programming model a process is an independently executing object with a process body. The process body is written in a subset of C syntax and uses intrinsic functions to perform stream or signal operations. A process may be either software or hardware. All declared processes are initiated when the program begins and runs until their subroutine bodies complete their tasks/functions.

In the following, the directives format is recalled for describing processes, streams and signals that a StreamsC program uses. These directives are embedded in specially formatted blocks. Each directive must be on one line and prepended by "///" followed by a keyword identifying the directive and optional parameter(s) [1].

The first set of directives describes the run function of a process. This is the body of code that gets executed when the associated process is initiated. The PROCESS FUN directive gives a name to the run function, input and output streams and signal parameters, followed by an optional parameter to be passed to the process when it is initiated. After the parameter, the body of the function appears as a normal C code, usually containing variable declarations, stream and/or signal communication, and computation. Finally, a keyword directive is used to mark the end of the run function. The format of the PROCESS FUN directive is shown in Figure 1.

The stream and signal names can be used within stream operations within the body of the process. The data type of stream or signal elements precedes the name of the stream or signal. StreamsC provides predefined unsigned and signed integer data types of stream or signal elements for selected bit lengths ranging from 1 to 64. The supported bit lengths are 1, 2, 4, 6, 8, 12, 16, 18, 20, 24, 32, 40, 48, 64, 128. A simple convention is used to name these predefined types. The signed types have the name **sc_int <bit length>**. The unsigned types have the name **sc_uint<bit length>**.

To describe a process to StreamsC, the PROCESS directive is used. A process has an associated run function and it is an SP (software process) or HP (hardware process) type. If omitted, SP is assumed. Figure 2 shows the format of the PROCESS directive.

The last directive CONNECT is used to connect processes via streams and signals. To connect two processes, the name of one process's stream or signal is associated with the name of another process's stream or signal. In Figure 3, the stream or signal formal parameter defined in the PROCESS FUN directive is generically referred to as a port. The CONNECT directive must be specified from "source" to "destination" (see Figure 3).

Note that the connections between processes must be one-to-one. Broadcast patterns and many-to-one connections are not supported.

An example of the use of these directives to declare and connect processes is shown in Figure 4. There are two software processes called

```
/// CONNECT <process_name>.<port> <process_name>.<port>
      Where: <port> ::= stream or signal name from a PROCESS_FUN directive
```

Figure 3. Format of a StreamsC CONNECT directive

```
//
// Process Functions definitions
//
/// PROCESS_FUN setup_run
/// OUT_STREAM sc_uint4 data
/// PROCESS FUN BODY
    ... C code ...
/// PROCESS_FUN_END
/// PROCESS_FUN finish_run
/// IN_STREAM sc_uint4 processed_data
/// PROCESS_FUN_BODY
    ... C code ...
/// PROCESS_FUN_END
/// PROCESS_FUN p_run
/// IN_STREAM sc_uint4 str1
/// OUT_STREAM sc_uint8 str2
/// PROCESS_FUN_BODY
    ... C code ...
/// PROCESS_FUN_END
//
// Process definitions
//
/// PROCESS setup PROCESS_FUN setup_run
/// PROCESS p_1 PROCESS_FUN p_run TYPE HP
/// PROCESS p_2 PROCESS_FUN p_run TYPE HP
/// PROCESS finish PROCESS_FUN finish_run
//
// Connections
//
/// CONNECT setup.data p_1.str1
/// CONNECT p_1.str2 p_2.str1
/// CONNECT p_2.str2 finish.processed_data
```

Figure 4. CONNECT directives example

setup and finish, and two hardware processes which are instances of the p process. The first instance of the p process (p_1) receives stream data from the setup process. The second instance of the p process (p_2) receives data from the previous instance and outputs data to the finish process.

4. STARSoC design tool

STARSoC [4] is a framework for hardware/software co-design, design space exploration and rapid prototyping on an FPGA (Field Programmable Gate Array) platform for Multi-Processor Systems on Chip (MPSoCs). The overall design flow of the STARSoC tool is summarized in Figure 5.

The design methodology in the STARSoC tool starts from a global model of an application which is a set of communicating processes described in the StreamsC textual language. In the StreamsC model, a process may be either a software process (SP) or a hardware process (HP). Software and hardware processes represent the software and hardware part of the system, respectively. The hardware and software partitions are defined by the user. Note that this design

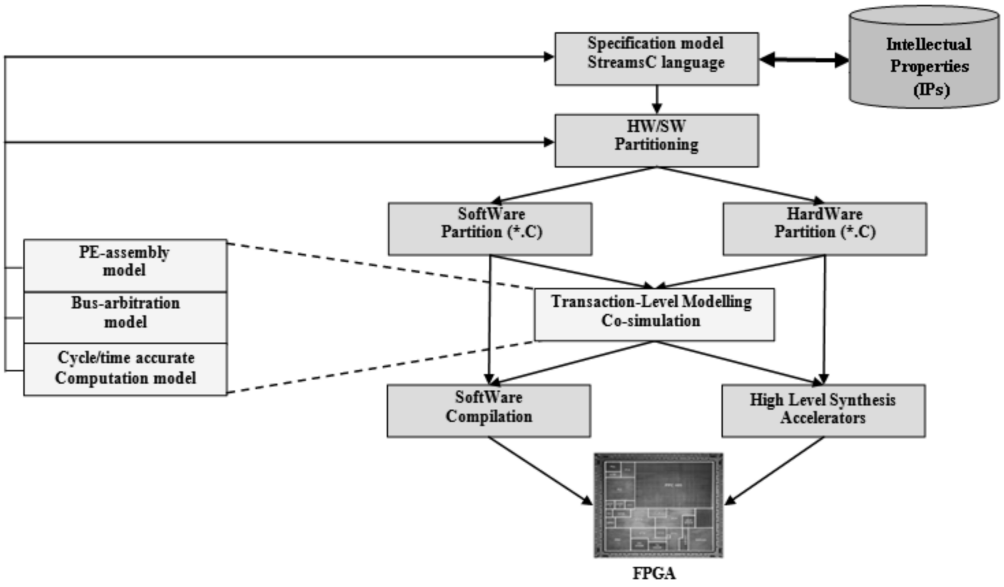


Figure 5. STARSoC design flow [21]

flow is based mainly on reusing open source Intellectual Properties (IPs) for both hardware and software parts.

After hardware-software partitioning, the hardware part is synthesized in Register-Transfer Level (RTL) re-using the StreamsC compiler [22]. In addition, the hardware interface allowing the two partitions, i.e. hardware and software, to communicate is also generated in the RTL code. The obtained RTL code is then downloaded to the FPGA. The software part will be compiled and re-instrumented to generate the machine code of the software processes. This machine code is then downloaded into the program memory of each available processor in the generated MPSoC platform. As a result, STARSoC generates a bus-based MPSoC platform from a high-level application specification.

Before building a prototype for an application, the STARSoC performs a hardware/software co-simulation to validate the behaviour for both hardware and software components and also the interaction between them. In addition, co-simulation permits the performance analysis and rapid exploration of several solutions containing different descriptions of the system components. For this purpose, The STARSoC tool uses Transaction-Level Modelling (TLM) framework [23] which is commonly used for the fast simulation and design exploration of a complex System on Chips (SoCs) at several levels of abstraction and detail. TLM proposes four well-defined transaction level abstraction models that can be independently validated, simulated and estimated. In these models, the application is represented as a set of communicating processes where the communication and the computation are explicitly separated. These processes perform computations and communicate with other processes through an abstract channel.

On the basis of the specification model which describes system functionality without any architecture details (obtained from process codes), the STARSoC tool performs co-simulation by using the following TLM model levels shown in Figure 5:

- *PE-assembly model:* it is made up with multiple processing elements (PEs) connected by channels.
- *Bus-arbitration model:* it represents a refined PE-assembly model in the communication part.
- *Cycle/time-accurate computation model:* It contains cycle accurate computation and approximate-timed communication. This model can be generated from the bus-arbitration model.

The advantage of this approach is that it allows designers to exploit the platform at the earlier stages of the design flow.

5. Eclipse modelling project – overview

The Eclipse Modelling Project [24] is a collection of frameworks and tools for the Model Driven Engineering on the Eclipse platform. In short, they provide a wide range of solutions for various aspects of model driven development, from language definition, generative development of language editors to code generation as well as model verification and validation [25]. In the following, some of the tools from Eclipse Modelling Project that have been used in this work are introduced. These tools are specifically recommended as a basis for developing a graphical editor for the STARSoC tool.

5.1. Eclipse Modelling Framework (EMF)

The Eclipse Modelling Framework [26] forms the basis for all Eclipse Modelling Project tools. It represents the modelling framework and the code generation facility for specifying meta-models and managing model instances. More precisely, EMF includes its own meta-modelling language called Ecore which is used for defining the abstract syntax of modelling languages [27]. From a modelling language specification defined by the Ecore meta-model, EMF generates a simple tree-based editor that enables viewing and editing the instances of the modelling language. In addition, EMF comes with a set of related frameworks for validating models, creating and executing queries against EMF models as well as model transactions.

5.2. Graphical Editing Framework (GEF)

Although EMF is able to generate tree-based editors for model instances of existing meta-models, these editors do not suffice since models are better rendered in a true graphical way. The Graphical Editing Framework [28] provides technology to aid developers in creating rich graphical editors, which are not easily built using native widgets found in the base Eclipse platform. It contains the entire set of tools to define a graphical concrete syntax for each entity of the meta-model according to its appropriate graphical notation. In addition, GEF employs a Model-View-Controller (MVC) architecture which is used to interconnect the graphical part of an editor with the model elements. Thereby, it permits changes to be applied to the model from the view [25]. Although EMF and GEF can be used separately, building a graphical editor requires both of them. In this sense, GEF provides the graphical support required for building a diagram editor on the top of the EMF framework.

5.3. Graphical Modelling Framework (GMF)

The Graphical Modelling Framework [29] provides a generative component and runtime infrastructure for developing graphical editors based on EMF and GEF. In other words, it provides a generative bridge between the EMF (that allows the meta-model definition) and GEF (a lightweight graphical framework, based on MVC architecture) to help developers creating enhanced graphical editors [25]. Using this framework, one can define graphical notations for existing EMF meta-models.

5.4. Acceleo language

Acceleo is a model-to-text transformation framework that generates text from models [30]. It has been in development since 2006, and was incorporated into the Eclipse M2T project in 2009 [24]. Its purpose is to implement code generators with an easy to use language (according to Object Management Group's MOF model to text transformation language standard [31]) and a good enough tool support (IDE, syntax highlighting, error reporting and debugging features). An Acceleo program requires a meta-model and a model compliant with this meta-model, from which it

generates a text or a code. The meta-model and the model are defined using the EMF framework, which makes Acceleo compatible with other tools based on EMF.

The Acceleo language is a template based approach wherein the text or code to be generated from models are specified as a set of text templates that are parameterized with model elements. More precisely, Acceleo scans the source model according to its meta-model and defines a textual template in the relevant syntax for each text fragment to be generated. The variable parts in the text fragment are specified over model elements. An advantage of this situation is the fact that the structure of the Acceleo templates will directly reflect the structure of the generated text. Thus, the destination text is directly generated, with no need for post-processing. The main feature of Acceleo is that the generated text is mixed with Acceleo syntax.

6. Graphical modelling editor for STARSoC

As it was mentioned earlier, the STARSoC tool starts from a StreamsC textual specification which consists of the architecture and behaviour of a complex embedded system. Gathering all system aspects in StreamsC textual specifications increase their complexity, decrease their readability, and make their understanding and maintenance more difficult. To remedy this, the authors propose to develop a graphical modelling editor for the STARSoC design tool which combines the architectural and behavioural aspects of the system in one model. The architectural aspect is expressed with a UML Component-like Diagram serving this purpose, whereas the behavioural aspect is specified in the C syntax. From this whole model, the StreamsC specification can be generated and all STARSoC design flow activities can be performed.

This section provides the outline of, the process of building the proposed graphical modelling editor using the well-known frameworks defined in MDE approach on the Eclipse platform. The presented approach consists of a process with two steps:

1. The first step consists of specializing UML Component Diagram [6] into StreamsC structural concepts. For this purpose a meta-model for the specialised UML Component Diagram is proposed and a graphical modelling editor is built according to the proposed meta-model.
2. The second step encompasses defining the code generation of StreamsC specification. In order to obtain the automatic and correct process of the code generation, the authors propose to use an Acceleo template language to define and implement the transformation.

6.1. Specializing UML Component Diagram into StreamsC structural concepts

To define a new modelling language or to extend and adapt an existing one, it is necessary to provide an abstract syntax (i.e. a meta-model denoting constructs, their attributes, relationships and constraints) as well as concrete graphical syntax information (the appearance of constructs and relationships in the graphical editor). In this work, the authors prefer to adapt an existing modelling language rather than to develop a new modelling language for specifying systems on the STARSoC tool.

Since StreamsC specification consists of a set of communicating parallel software and hardware processes described with a high level textual language and each process may be linked to a connector by an input port or an output port, the authors propose a modelling language adapted from UML Component Diagram [6] which meets additional needs for specifying embedded systems. UML Component Diagrams are widely used to define the architecture and the structure of a system. A Component Diagram shows components, their contents (source code, binary code or executable one), required interfaces, ports and relationships between them. For this purpose, the authors proposed to meta-model the structural aspect of StreamsC language expressed in the UML Component-like Diagram with the meta-model

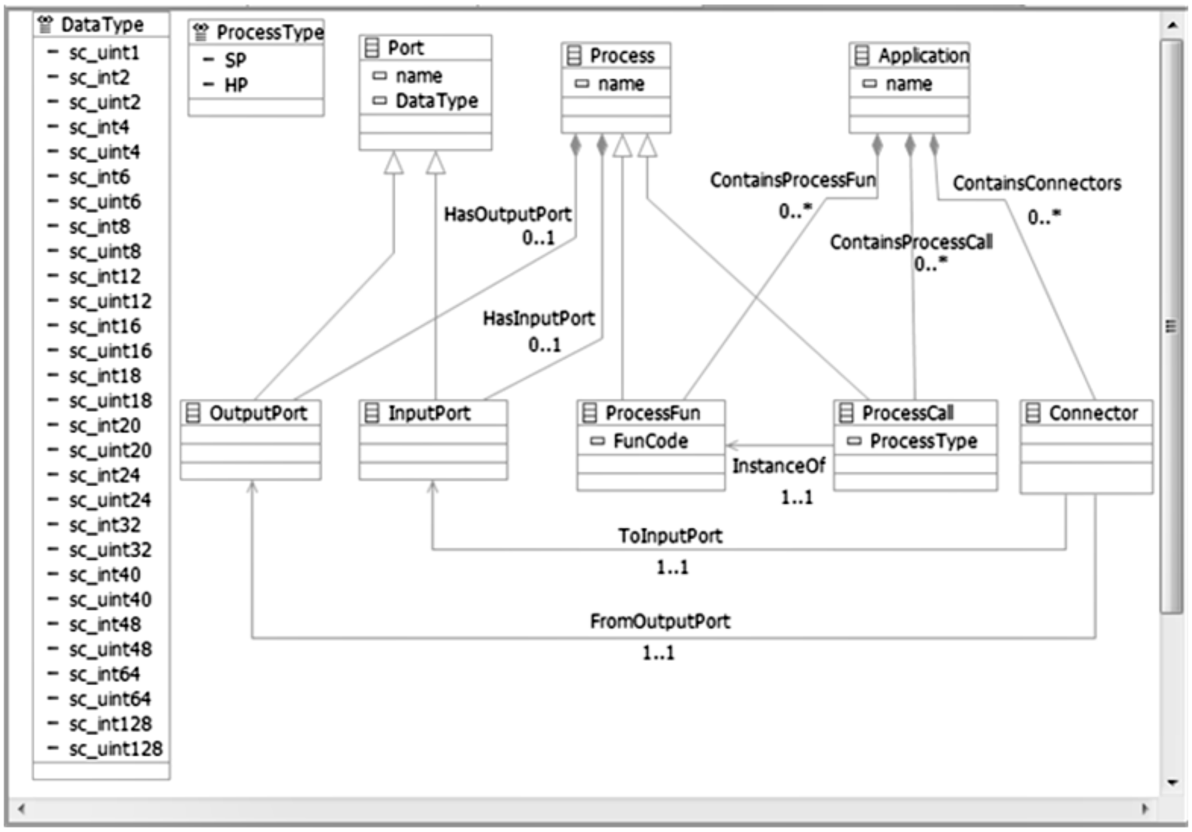


Figure 6. Proposed meta-model in Ecore

shown in Figure 6. In EMF, a meta-model is created and defined in the Ecore format, which is basically a sub-set of UML Class Diagrams. The proposed Ecore model is composed by the following classes:

The *Application* class (attribute name: *name*) represents the application. It contains all the elements used in the application which are process function definitions (*ProcessFun*), process definitions (*ProcessCall*) and connections between processes (*Connector*). The containment relations between the Application class and these elements are specified with Composition relations as shown in Figure 6.

The *ProcessFun* class represents the run functions of processes. It has a String attribute named *FunCode* containing the function code that gets executed when the associated process is initiated.

The *ProcessCall* class represents initiated processes in the application. Each process has an associated run function which is specified with an *Instanceof* association, and a *ProcessType* attribute to indicate the type of the process. The *ProcessType* attribute takes its value from *ProcessType* enumeration class which is *SP* (Software Process) or *HP* (Hardware Process).

The *Connector* class represents the connections between processes via Ports. A connector has two associations with two other classes called *OutputPort* and *InputPort*, which are sub-classes of the *Port* class.

The *OutputPort* class describes the output ports of source processes, whereas the *InputPort* class represents the input ports of destination processes. *OutputPort* and *InputPort* classes inherit two attributes from the *Port* class: the name of the port and the data type of the stream or signal which takes its value from the *DataType* enumeration class.

In addition, *OutputPort* and *InputPort* classes are contained in the *Process* class which is the abstract class of *ProcessFun* and *ProcessCall* classes.

Despite its expressiveness, Ecore cannot cover all modelling constraints for a modelling language using only graphical elements. Usually, OCL is

```
import ecore : 'http://www.eclipse.org/emf/2002/Ecore#/';

package STARSoC : STARSoC = 'http://STARSoC/'
{
    class Application
    {
        attribute name : String[?] = 'MyApplication';
        property ContainsConnectors : Connector[*] { composes };
        property ContainsProcessCall : ProcessCall[*] { composes };
        property ContainsProcessFun : ProcessFun[*] { composes };
    }
    class Connector
    {
        invariant Connector_Must_Connect_Two_Ports_of_The_Same_DataType:
        self.ToInputPort.DataType = self.FromOutputPort.DataType;
        property ToInputPort : InputPort[?];
        property FromOutputPort : OutputPort[?];
    }
    class ProcessCall extends Process
    {
        invariant ProcessCall_InPort_DataType_Is_As_ProcessFun_InPort_DataType:
        self.HasInputPort.DataType = self.InstanceOf.HasInputPort.DataType;
        invariant ProcessCall_outPort_DataType_Is_As_ProcessFun_outPort_DataType:
        self.HasOutputPort.DataType = self.InstanceOf.HasOutputPort.DataType;
        attribute type : ProcessType[?];
        property InstanceOf : ProcessFun[1];
    }
    class ProcessFun extends Process
    class Process
    class Port
    class InputPort extends Port;
    class OutputPort extends Port;
    enum ProcessType
    enum DataType
}
```

Figure 7. Corresponding OCL invariants of the rules

employed to define additional constraints as the so-called well-formedness rules. These rules are implemented in OCL as invariants which are attached to meta-model classes in order to describe properties that should always be satisfied for every model. Thus, the invariant constraints are defined on the meta-model and validated on the model level using the EMF Validation Framework [32]. By introducing the OCL invariants for meta-model classes, a modelling language is more precisely defined leading to models with higher quality.

For this purpose, the proposed Ecore model was enriched with three OCL invariant constraints. These invariants allow the user to check the correctness of the described models with respect to their construction rules as stated in the StreamsC language. In the following part, these rules are described in a natural language, and subsequently the corresponding OCL invariants in the OCLinEcore text editor [33], which embeds the OCL expressions directly into Ecore models by annotating the relevant classes, are shown in Figure 7.

Rule 1: The two end ports of the *Connector* must have the same data type to assure their compatibility.

Rule 2: *ProcessCall* must have the data type of the input port as declared in the input port of its corresponding *ProcessFun*.

Rule 3: *ProcessCall* must have the data type of the output port as declared in the output port of its corresponding *ProcessFun*.

EMF from the proposed Ecore model was used to generate a simple tree-based editor for the modelling language that enables editing and viewing model instances. To develop its graphical modelling editors, both GEF and GMF were used

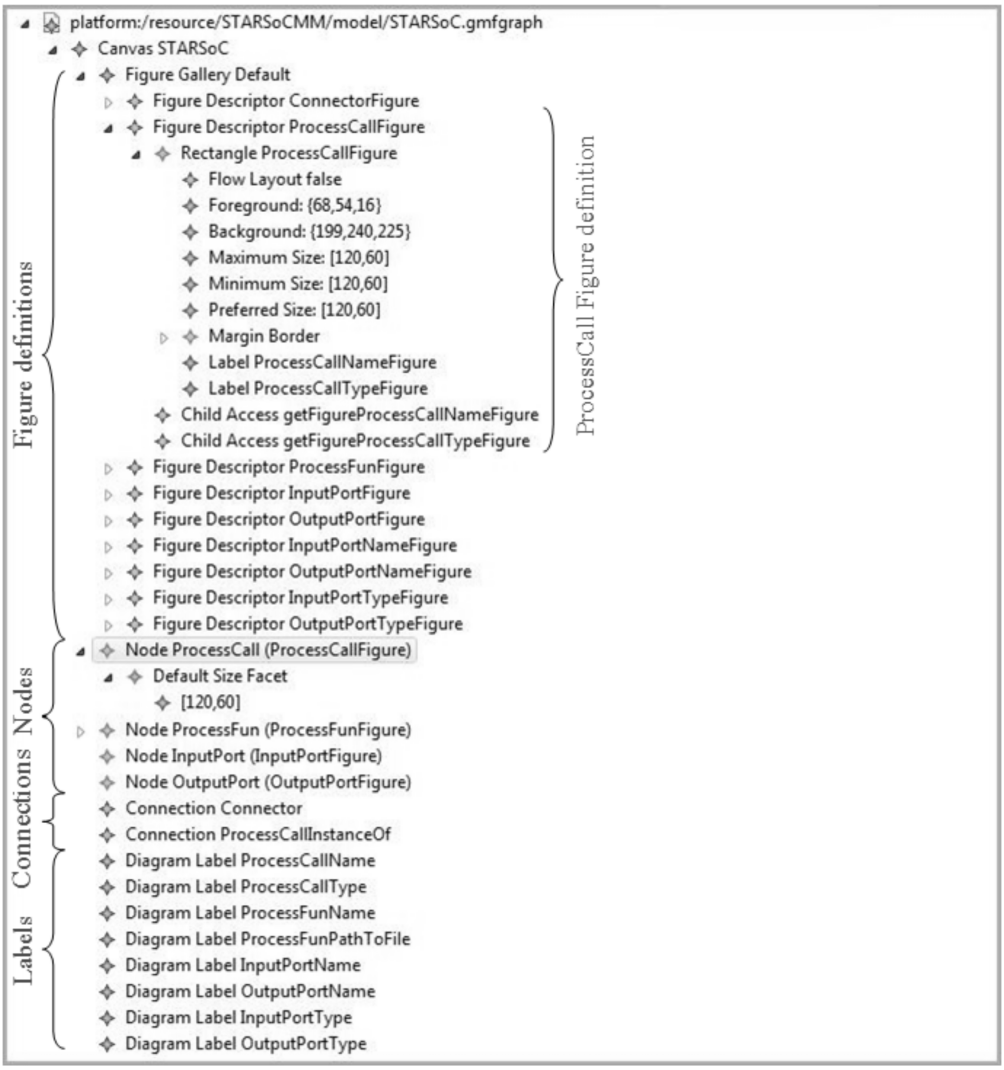


Figure 8. Graphical concrete syntax definition

to define the Graphical model and the Tooling model and Mapping model, respectively.

The Graphical model defines the concrete syntax of the modelling language according to their appropriate graphical notations. It includes information related to the graphical elements (i.e. nodes, labels, connections and decorations for connection ends) that will appear in the editor. The Graphical model contains also a *Figure Gallery* that contains figures which are used to define shapes. The elements that define the nodes, connections and labels are under the *Figure Gallery* root in the graphical model. Figure 8 shows the graphical definition model for the proposed Ecore model. For example, the *ProcessCall* node uses the rectangle shape defined under *ProcessCallFigure* Figure Descriptor. The rectangle sizes, colours, borders and labels are described separately as rectangle attributes. Similarly, each node element of the Ecore model references the corresponding Figure Descriptor.

The Tooling model defines the toolbar, menus to be used and other periphery to facilitate the management of the model content in the editor. The main focus of the Tooling model is the toolbar definition. The toolbar is defined within a Palette and contains Tool Groups which contain the Tools. In Figure 9, the Tooling definition model for this editor consists of three Tool Groups, namely *Processes*, *Ports* and *Connectors*. The *Processes* Tool Group contains the *ProcessFun* and *ProcessCall* tools for creating the

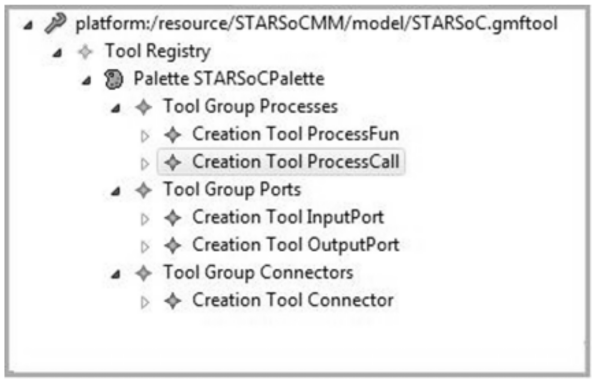


Figure 9. Tooling definition model

Figure 10. Mapping definition model

ProcessFun and *ProcessCall* elements. The *Ports* Tool Group includes *InputPort* and *OutputPort* tools for creating the Input Port and Output Port elements. The last tool group concerns the creation of *Connectors* in the models.

The Mapping model maps graphical elements from the graphical definition model and creation tools from the tooling definition model to the language constructs from the meta-model. The Mapping model consists of several *Top Node References*, each of which contain one *Node Mapping*. The *Node Mapping* is used to map an element in the graphical model to both the construct in the meta-model and to the creation tool. In addition, it is within the *Node Mapping* that the *Label Mappings* and *Child References* are defined. *Label Mappings* map a *Diagram Label* in the graphical model to an attribute in the meta-model class that is referenced by the enclosing *Node Mapping*. *Child References* allow meta-model elements to have children, where each child contains an inner *Node Mapping*. In addition to *Top Node References*, *Link Mapping* is used to specify information about a link. It contains information about a source feature, target feature, graphical representation, creation tool, and many other properties. For instance, according to the mapping model in Figure 10, *ProcessCall* elements (Fig. 6) are created by means of the Creation Tool *ProcessCall* (Fig. 9) and the graphical representation for them is the *ProcessCall* Figure definition (Fig. 8). For each *ProcessCall* the corresponding *"name"* and *"ProcessType"* attributes are also visualized because of the specified *Feature Label Mappings* which relate the attribute *"name"* (resp. the attribute *"ProcessType"*) of the *ProcessCall* class with the diagram label *ProcessCallName* (resp. *ProcessCallProcessType*) defined in the graphical definition model.

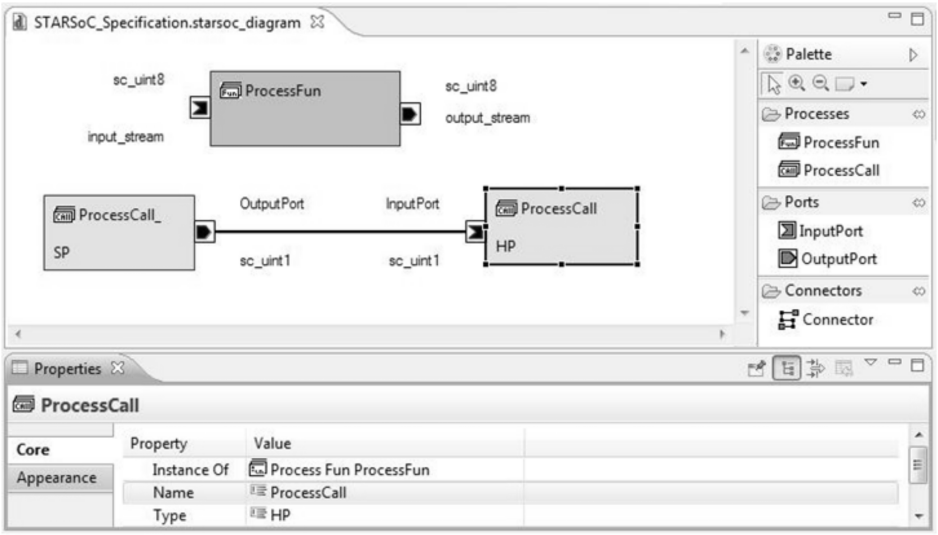


Figure 11. Generated editor for STARSoC

Finally, the Mapping model is transformed into a diagram generator model from which a diagram editor can be generated. Figure 11 shows the graphical modelling editor generated form EMF and GMF models defined for specifying systems on the STARSoC design tool. The editor shows the graphical elements in the diagram and the tools in the palette. Furthermore, GMF provides more advanced features such as annotating, zooming and layouting for the generated editor. The properties of a graphical element can be accessed through the properties view.

6.2. Code generation of StreamsC specification

The next step is the transformation of the graphical specification of a system into its equivalent StreamsC specification using the Acceleo transformation language. In order to do that, the preceding transformation was composed with a set of Acceleo templates (see Figure 12) that traverses the elements of the source model (instances of meta-models) and generates the corresponding StreamsC code.

The first Acceleo template *ToStreamsC (App : Application)* is the main template. It creates the file of the StreamsC specification and takes the only instance of the Application class which contains all model elements as a parameter. Using this parameter (*App*), it scans the contained elements and for each element type produces the corresponding StreamsC code. To achieve this, the *ToStreamsC* template uses three others templates defined for the ProcessFun, ProcessCall and Connector meta-model elements. For example, the template *GenProcessFun(pf : ProcessFun)* takes ProcessFun *pf* as a parameter and writes the run function description of *pf*, which contains the PROCESS_FUN directive, the name of the run function, the input and the output streams, the body of the function and the PROCESS_FUN_END directive, to the output file.

7. Case study

To evaluate the practical usefulness of the proposed graphical editor, a simple application of image processing involving the horizontal edge detection of an image of 256 X 256 pixels coded out of 8 bits was considered. The edge detection is a preliminary step in most image processing techniques. Figure 13 presents the model created in this editor.

The application is defined through two different processes. The first one is a software process, it allows to send the original image, through its output stream, in the direction of the input stream of the second process which is a hardware process. The hardware process performs edge

```
[comment encoding UTF-8 /]
[module generate ('http://STARSoc/')]
[template public ToStreamsC (App : Application)]
   [comment @main/]
   [file (App.name.concat('.sc'), false, 'UTF-8')
   /*______[App.name/].sc_____________________________________________
       Automatically generated streamsc specification
   ___________________________________________________________________*/
   //
   // Process Functions definitions
   //
   [for (processFun : ProcessFun| App.ContainsProcessFun)]
   [GenProcessFun(processFun) /]
   [/for]
   //
   // Process definitions
   //
   [for (processCall : ProcessCall| App.ContainsProcessCall)]
   [GenProcessFun(processCall)/]
   [/for]
   //
   // Connections
   //
   [for (connector : Connector| App.ContainsConnectors)]
   [GenConnector(connector, App)/]
   [/for]
   [/file]
[/template]

[template private GenProcessFun(pf : ProcessFun)]
   /// PROCESS_FUN [pf.name/]
   /// IN_STREAM [pf.HasInputPort.DataType/] [pf.HasInputPort.name/
   /// OUT_STREAM [pf.HasOutput Port.DataType/] [pf.HasOutputPort.name/]
   /// PROCESS_FUN_BODY
   [pf.FunCode /]
   /// PROCESS_FUN_END
[/template]

[template private GenProcessCall(pc : ProcessCall)]
   /// PRoCESS [pc.name/] PROCESS_FUN [pc.InstanceOf.name/] TYPE [pc.ProcessType/]
[/template]

[template private GenConnector(c : Connector, App:Application)]
   /// CONNECT
   [for (pc :ProcessCall| App.ContainsProcessCall)]
      [if(pc.HasOutputPort=c.FromOutputPort)][pc.name/][/if]
   [/for]
   .[c.FromOutputPort.name/]
   [for (pc :ProcessCall| App.ContainsProcessCall)]
      [if (pc.HasInputPort=c.ToInputPort)][pc.name/][/if]
   [/for]
   .[c.ToInputPort.name/]
[/template]
```

Figure 12. Acceleo templates for StreamsC code generation

Figure 13. Graphical specification of the application

detection image and returns the resulting image to the software process.

The edge detection algorithm calculates the absolute value of the difference between two consecutive pixels arriving on the data bus of the input streams. The equation of the horizontal edge detection filter is as follows:

$$y(x) = |x(n) - x(n-1)| \tag{1}$$

Only one hardware process is sufficient to perform this calculation. The algorithm of horizontal edge detection is described below (see Figure 14).

Its equivalent StreamsC description is generated from the graphical specification of the application. To generate StreamsC specification in this approach, it is necessary to execute the Acceleo template defined in the previous section. The automatic generated file Essai.sc, which contains the specification, is shown in Figure 15.

This StreamsC specification of the whole application is the basis on which all STARSoC design activities can be performed. Figure 16 shows the development environment for STARSoC tool.

8. Conclusion

The paper presents some attempts to improve the STARSoC design tool by taking advantage of the Model Driven Engineering techniques. More precisely, Eclipse Modelling Project frameworks and tools (EMF, GEF, GMF, Acceleo,...), which follow the principles of MDE approach, were used to develop a graphical editor for the STARSoC design tool. This editor supports the graphical editing of embedded system models in terms of UML Component-like Diagram and generates the StreamsC textual specifications of these models. The adapted UML Component Diagram is defined in accordance with the embedded system design needs using the Ecore model, whereas the transformation process is defined and executed using the Acceleo framework. The resulting StreamsC specifications are used to perform all STARSoC design tool activities, such as hardware/software co-design, design space exploration and high level synthesis.

According to the authors this approach is sufficiently flexible to incorporate new design needs. Due to the employed Eclipse Modelling Project, revisions of the meta-model almost automatically yield an updated editor and the generation of a text or code is supported as the coding of each meta-model element is analysed separately.

Future work plans encompass the use and adaptation of some UML behavioural diagrams in order to depict the behavioural features of embedded system processes. These behavioural diagrams will be used to automatically generate process codes. One promising direction is to combine existing UML profiles for embedded systems design, such as SysML and MARTE profiles. This

```
sc_uint8 data_in, data_out, x, y;
sc_stream_open(input_stream);
sc_stream_open(output_stream);
while (!sc_stream - eos(input_stream)) {
#pragma SC pipeline
data_in = sc_stream_read(input_stream);
sc_stream_write(output_stream.data_out);
y = x - data_in;
x = data_in;
If (y >= 0) { data_out = y; }
Else { data_out = y * (-1); }
}
sc_stream_close(input_stream);
sc_stream_close(output_stream);
```

Figure 14. Horizontal edges detection algorithm

Figure 15. Generated StreamsC specification

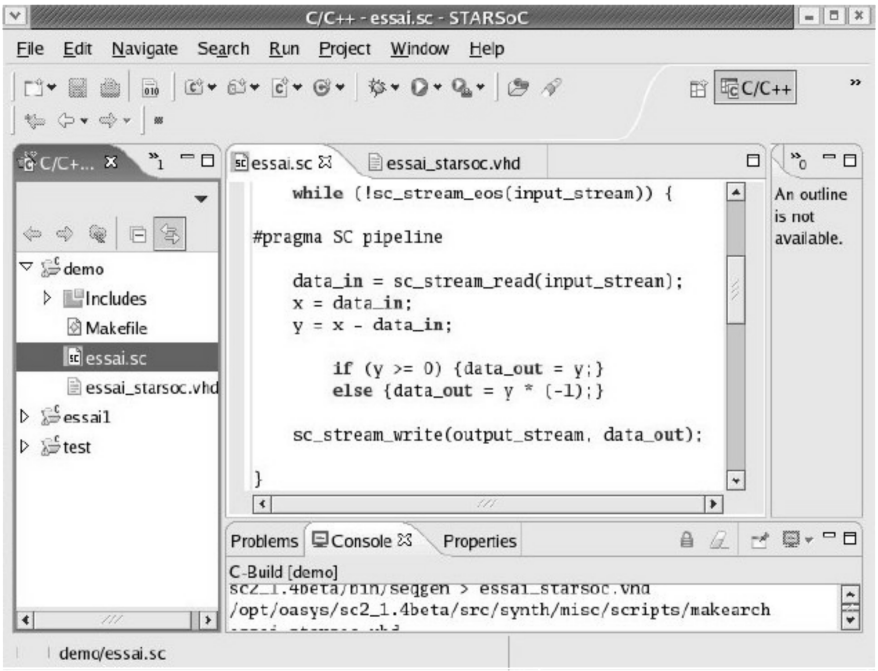


Figure 16. The development environment for STARSoC tool

combination is possible since most of the profiles are focused on the process paradigm.

References

[1] M. Gokhale, *sc2 Reference Manual*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2003.

[2] W. Meeus, K.V. Beeck, T. Goedemé, J. Meel, and D. Stroobandt, "An overview of today's high-level synthesis tools," *Design Automation for Embedded Systems*, Vol. 16, No. 3, 2012, pp. 31–51.

[3] J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K.A. Vissers, and Z. Zhang, "High-level synthesis for FPGAs: From prototyping to deployment," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 30, No. 4, 2011, pp. 473–491.

[4] A. Samahi and E. Bourennane, "Automated integration and communication synthesis of reconfigurable MPSoC platform," in *Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, University of Edinburgh. Scotland, United Kingdom: IEEE Computer Society, 2007, pp. 379–385.

[5] M. B.Gokhale, J.M. Stone, J. Arnold, and M. Kalinowski, "Stream-oriented FPGA computing in the Streams-C high level language," in *Proceedings of the 2000 IEEE Symposium on Field-Programmable Custom Computing Machines*. Napa Valley, CA, USA: IEEE Computer Society, 2000, pp. 49–56.

[6] *Unified Modeling Language, Version 2.5*, Object Management Group, 2015, OMG Document Number: formal/15-03-01. [Online]. http://www.omg.org/spec/UML/2.5/PDF

[7] A.R. da Silva, "Model-driven engineering," *Computer Languages, Systems and Structures*, Vol. 43, No. C, 2015, pp. 139–155.

[8] J. Joven, O. Font-Bach, D. Castells-Rufas, R. Martínez, L. Terés, and J. Carrabina, "xENoC – an experimental network-on-chip environment for parallel distributed computing on NoC-based MPSoC architectures," in *16th Euromicro International Conference on Parallel, Distributed and Network-Based Processing*. Toulouse, France: IEEE Computer Society, 2008, pp. 141–148.

[9] D. Thomas and P. Moorby, *The Verilog Hardware Description Language*, 3rd ed. Norwell, MA, USA: Kluwer Academic Publishers, 1996.

[10] J. Keinert, M. Streubuhr, T. Schlichter, J. Falk, J. Gladigau, C. Haubelt, J. Teich, and M. Meredith, "SystemCoDesigner – an automatic ESL synthesis approach by design space exploration and behavioral synthesis for streaming applications," *ACM Transactions on Design Automation of Electronic Systems*, Vol. 14, No. 1, 2009, pp. 1–23.

[11] T. Grotker, *System Design with SystemC*. Norwell, MA, USA: Kluwer Academic Publishers, 2002.

[12] *SOPC Builder User Guide, Version 1.0*, Altera Corporation, San Jose, CA, USA, 2010, Document Number: UG-01096. [Online]. http://www.altera.com/literature/ug/ug_SOPC_builder.pdf

[13] *EDK Concepts, Tools, and Techniques: A Hands-On Guide to Effective Embedded System Design, Version 13.2*, Xilinx Online Documents, 2011, OMG Document Number: UG683. [Online]. http://www.xilinx.com/support/documentation/sw_manuals/xilinx13_2/edk_ctt.pdf

[14] *Systems Modeling Language (OMG SysML), Version 1.4*, Object Management Group, 2015, OMG Document Number: formal/2015-06-03. [Online]. http://www.omg.org/spec/SysML/1.4/

[15] *A UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded systems, Version Beta 2*, Object Management Group, 2008, OMG Document Number: ptc/2008-06-09. [Online]. http://www.omg.org/omgmarte/Documents/Specifications/08-06-09.pdf

[16] *DaRT team: Dataparallelism for Real-Time*. [Online]. http://www.inria.fr/en/teams/dart/ [Accessed 2016].

[17] *GASPARD2 SoC Framework*. [Online]. http://www.gaspard2.org/ [Accessed 2016].

[18] *Model Driven Architecture Guide, Version 1.0*, Object Management Group, 2003, OMG Document Number: omg/2003-05-01. [Online]. http://www.omg.org/mda/mda_files/MDA_Guide_Version1-0.pdf

[19] *UML Profile for System on a Chip (SoC), Version 1.0.1*, Object Management Group, 2006, OMG Document Number: formal/2006-08-01. [Online]. http://www.omg.org/spec/SoCP/1.0.1/PDF

[20] T. Kangas, P. Kukkala, H. Orsila, E. Salminen, M. Hännikäinen, T.D. Hämäläinen, J. Riihimäki, and K. Kuusilinna, "UML-based multiprocessor SoC design framework," *ACM Transactions on Embedded Computing Systems*, Vol. 5, No. 2, 2006, pp. 281–320.

[21] S. Boukhechem and E. Bourennane, "SystemC transaction-level modeling of an MPSoC platform based on an open source ISS by using interprocess communication," *International Journal of Reconfigurable Computing*, Vol. 2008, 2008, pp. 1–10.

[22] J. Frigo, *sc2 Hardware Library Reference Manual*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2000.

[23] L. Cai and D. Gajski, "Transaction level modeling: An overview," in *Proceedings of the 1st IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*. Newport Beach, CA, USA: ACM, 2003, pp. 19–24.

[24] *Eclipse Modelling Project (EMP)*. [Online]. http://www.eclipse.org/modeling/ [Accessed 2016].

[25] R.C. Gronback, *Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit*, 1st ed. Addison-Wesley Professional, 2009.

[26] *Eclipse Modelling Framework (EMF)*. [Online]. https://eclipse.org/modeling/emf/ [Accessed 2016].

[27] D. Steinberg, F. Budinsky, M. Paternostro, and E. Merks, *EMF: Eclipse Modeling Framework 2.0*, 2nd ed. Addison-Wesley Professional, 2009.

[28] *Graphical Editing Framework (GEF)*. [Online]. http://www.eclipse.org/gef/ [Accessed 2016].

[29] *Graphical Modelling Framework (GMF))*. [Online]. http://www.eclipse.org/modeling/gmp/ [Accessed 2016].

[30] *User Guide, Version 3.1.0*, The Eclipse Foundation, 2011. [Online]. http://www.eclipse.org/acceleo/support/

[31] *MOF Model to Text Transformation Language, Version 1.0*, Object Management Group, 2008, OMG Document Number: formal/2008-01-16. [Online]. http://www.omg.org/spec/MOFM2T/

[32] *The EMF Validation Framework project (EMF-VF)*. [Online]. http://www.eclipse.org/modeling/emf/?project=validation [Accessed 2016].

[33] *OCLinEcore Editor*. [Online]. https://wiki.eclipse.org/MDT/OCLinEcore [Accessed 2016].

e-Informatica Software Engineering Journal, Volume 12, Issue 1, 2018, pages: 27–49, DOI 10.5277/e-Inf180102

An Empirical Study on the Factors Affecting Software Development Productivity

Luigi Lavazza*, Sandro Morasca*, Davide Tosi*
*Dipartimento di Scienze Teoriche e Applicate, Università degli Studi dell'Insubria
luigi.lavazza@uninsubria.it, sandro.morasca@uninsubria.it, davide.tosi@uninsubria.it

Abstract

Background: Software development productivity is widely investigated in the Software Engineering literature. However, continuously updated evidence on productivity is constantly needed, due to the rapid evolution of software development techniques and methods, and also the regular improvement in the use of the existing ones.
Objectives: The main goal of this paper is to investigate which factors affect productivity. It was also investigated whether economies or diseconomies of scale exist and whether they may be influenced by productivity factors.
Method: An empirical investigation was carried out using a dataset available at the software project repository ISBSG. The major focus was on factors that may affect productivity from a functional point of view. The the conducted analysis was compared with the productivity data provided by Capers Jones in 1996 and 2013 and with an investigation on open-source software by Delorey et al.
Results: This empirical study led to the discovery of interesting models that show how the different factors do (or do not) affect productivity. It was also found out that some factors appear to allow for economies of scale, while others appear to cause diseconomies of scale.
Conclusions: This paper provides some more evidence about how four factors, i.e., programming languages, business areas, architectural types, and the usage of CASE tools, influence productivity and highlights some interesting divergences in comparison with the results reported by Capers Jones and Delorey et al.

Keywords: effort, function point, empirical study, ISBSG dataset, factors, development, productivity

1. Introduction

Productivity is one of the crucial aspects in software development, as it is intrinsically related to software costs. Improvements in software development productivity may come from the industrial use of novel techniques constantly introduced in Software Engineering. Also, software development productivity may improve because of the ever increasing knowledge and experience acquired on existing software engineering techniques, which, in addition, are becoming more and more consolidated over time. However, it needs to be checked if this potential improvement in productivity actually takes place and, if so, to what extent and under what conditions, so that conditions favoring productivity can be created and maintained in the software industry.

Many factors are believed to significantly influence productivity [1], so identifying relationships between factors and productivity is no simple matter. In addition, Software Engineering is still a relatively recent discipline and its empirical laws still need to be accurately described and validated. Moreover, Software Engineering is very human-intensive, thus productivity is certainly affected by factors that may not be easy to quantify and control. The human-intensive nature of

software development may also imply that there are intrinsic limits to potential improvements in productivity.

This paper reports on an empirical study which was carried out to investigate whether and to what extent productivity is influenced by a number of factors, namely, the primary programming language used to develop each software project, the business area addressed by the project, the architectural type adopted by the project, and the use of CASE (Computer-Aided Software Engineering) tools. It was also investigated whether economies or diseconomies of scale (i.e. the cost disadvantages that companies accrue due to an increase in company size or output resulting in the production of services at increased per-unit costs) may exist and whether they depend on the factors that influence productivity.

The data used in this empirical study came from projects in the ISBSG (International Software Benchmarking Standards Group)[1] dataset [2], one of the most extensive datasets containing data on software development projects, and especially effort data, spanning 25 years. The ISBSG dataset contains data from a few thousand projects. Even though this is a fairly large amount of data, the ISBSG dataset represents a limited sample of the software development projects that have been and still are being carried out worldwide. Moreover, its data are provided on a voluntary basis by different types of software developers. As a result, ISBSG data may be only partially representative of all current software development practices. At any rate, ISBSG data are about projects with the same or similar characteristics as a fairly large part of current software development projects.

The main focus was on productivity from a functional point of view, so the functional size of product is measured (in Function Points [3,4]), rather than the physical size (e.g. measured in Lines of Code – LoC).

The set of factors investigated in this paper extends the set of factors studied in the authors' previous work [5], in which they were only interested in understanding the effect of the primary programming language on software productivity. In the work documented in this paper, more factors are investigated, as described in the following research question.

RQ1: Which factors influence productivity? Specifically: Does the primary programming language factor affect productivity (i.e. does productivity increase or decrease with the adopted programming language)? Does the business area factor affect productivity? Does the type of architecture factor affect productivity? Does the use of CASE tools affect productivity?

Also, the following additional research question, related to whether a factor may determine software development economies or diseconomies of scale, are addressed here.

RQ2: Which factors influence economies and diseconomies of scale? Specifically: Does the choice of the primary programming language determine a relation between size and development effort characterized by economies (or diseconomies) of scale? Similarly, do the business area, the type of architecture or the use of CASE tools determine a relation between size and development effort characterized by economies (or diseconomies) of scale?

Several different analyses were carried out. Firs a "naïve" analysis was carried out, by looking at the mean, median, and variance of the productivity for the projects in the ISBSG dataset and assessing differences across different subsets of projects, grouped according to the programming language and the other factors mentioned above. Then the productivity level of each programming language was compared with the data reported by Capers Jones [6, 7] and Delorey et al. [8], to investigate whether our productivity data are aligned with these reference data. To investigate the existence of economies and diseconomies of scale, regression models that correlate size and effort for each value of a productivity-influencing factor were built to highlight the dependence of productivity on size [1, 9] (see Section 7). All of the analyses done in the paper address both the complete ISBSG

[1] Most of the Repository Field Descriptions of the ISBSG dataset are available at: http://isbsg.org/2016/04/06/what-you-can-find-in-the-2016-r1-isbsg-development-enhancement-repository/.

data set and the "new development" and "enhancement" projects subsets separately.

The main contributions of our work with respect to the existing literature mainly lie in the fact that our study:

- is based on the analysis of a large, public dataset, namely the ISBSG dataset;
- provides up-to-date indications by analyzing recent software project data;
- addresses several factors that are believed to affect productivity;
- uses a rigorous statistical approach.

The remainder of the paper is organized as follows: Section 2 describes the analysis method used. Sections 3–6 report the analysis of productivity versus the considered factors. Section 7 discusses how each factor may contribute to the software development of economies of scale or diseconomies in software development. Section 8 lists possible threats to the validity of this work. Section 9 reviews related work. The conclusion are presented in Section 10.

2. Analysis method

2.1. Software development productivity

The adopted definition of productivity was very simple: the functional size of software developed divided by the amount of effort employed in the development process.

$$\text{Productivity} = \frac{\text{Size of developed software}}{\text{Software development effort}}$$

In this paper, the preferred size measures are the functional ones, mainly the Unadjusted Function Points (UFP) [4], although occasionally the lines of code (LoC) were used to compare these findings with those of other authors who used LoC measures. The amount of effort spent on developing software is given by the total number of person-hours or person-months spent in the development process.

2.2. The ISBSG dataset

The study reported here is based on the analysis of data from the ISBSG dataset release R12 [2]. The ISBSG dataset supports the definition of productivity given above. Specifically, many of the projects in the ISBSG dataset were measured by means of IFPUG (International Function Point Users Group) Function Points [4] or other essentially equivalent functional size measures, like NESMA (Netherlands Software Metrics users Association) Function Points [10]. The ISBSG dataset also contains development effort data, normalized to take into account possible differences in development processes.

The ISBSG dataset provides several product and process measures and characteristics that can be useful in a productivity study [11]. Among these, the programming language, the business area, the architecture and the usage of CASE tools are considered and analysed as factors that may affect productivity in this paper.

To study the effect of these factors on productivity, the authors selected and grouped data samples concerning projects with the same programming language, business area, architecture, or decision whether to use CASE tools.

2.2.1. New developments vs. enhancements

The ISBSG dataset contains data concerning both new developments and enhancements of software projects. To deal with enhancements, it is necessary to take into account the following issues.

- The size of an enhancement is defined differently than the size of development from scratch, as their measurement processes are different [12].
- The size of an enhancement in Function Points actually measures the size of the part of application in which the change occurs, not the size of the change [4, 12]. For instance, the introduction ofa new transaction has the same size as making a small change in an existing transaction, provided that the two transactions have the same complexity.
- A model stating that *Effort = f(functional size of the enhancement)* is, therefore, a simplification since enhancement effort depends on both the size of the change and the overall size of the product being changed. For

Table 1. New development projects from the ISBSG dataset: descriptive statistics

	Size [UFP]	Effort [PH]	Productivity [UFP/PH]
Mean	616	6766	0.176
Median	322	3226	0.110
Stdev	776	10497	0.253
Min	51	320	0.006
Max	7400	134211	3.960

instance, after an enhancement, a system test must be carried out, and the effort required for this type of testing is related to the entire application size, rather than the size of the enhancement alone. Unfortunately, building a model of the *Effort=f(functional size of the application, functional size of the enhancement)* type is not possible, since the ISBSG database does not provide the sizes of the enhanced applications,only the size of the enhancements.

Because of the differences in the development from scratch and enhancement processes, the effects of programming languages, business areas, architectural types, and usage of CASE tools are investigated on new developments and enhancement projects separately.

2.2.2. Data selection

Not all ISBSG projects were suitable for this analysis. Data samples were selected according to the following criteria:

- Only projects measured in IFPUG or NESMA FP and provided with both size and effort data were selected.
- Only data concerning projects with a specified primary programming language, business area, architecture, and usage of CASE tools were selected.
- The projects in the ISBSG dataset are characterized by different quality levels. The selected projects had their data quality rated 'A' or 'B', i.e., those with good quality of data in the ISBSG dataset. Similarly, the UFP rating (i.e. quality of functional size measurement) of the selected projects was 'C' or greater. This is consistent with the previous studies of the ISBSG dataset.
- New development projects concerning applications smaller than 50 UFP were not considered. For such small projects, it is likely that specific effects – such as the usage of simplified life cycles – can dramatically affect productivity, thus making them hardly comparable with larger projects.
- Similarly, projects greater than 10,000 UFP or requiring more than 150,000 person-hours were not considered. There were only 4 projects with such characteristics, so they can very well be considered outliers.

Table 2. Enhancement projects from the ISBSG dataset: descriptive statistics

	Size [UFP]	Effort [PH]	Productivity [UFP/PH]
Mean	293.7	4073.5	0.1
Median	167.5	2188	0.079
Stdev	403.7	6479.6	0.1
Min	50	322	0.004
Max	7134	109271	1.5

2.3. Descriptive statistics

2.3.1. New development projects

Out of about 6000 ISBSG projects, 989 data points concerning new developments satisfy the selection requirements described in Section 2.2.2. The descriptive statistics are given in Table 1 (where PH indicates person-hours).

Figure 1 shows the distribution of the productivity data of the selected projects (the grey diamond is the mean value). Projects with productivity greater than 1 UFP/PH are not shown, to preserve the readability of the figure.

2.3.2. Enhancement projects

Out of about 6,000 ISBSG projects, 1570 data points concerning enhancements satisfy the selection requirements described in the previous section. The descriptive statistics are presented in Table 2.

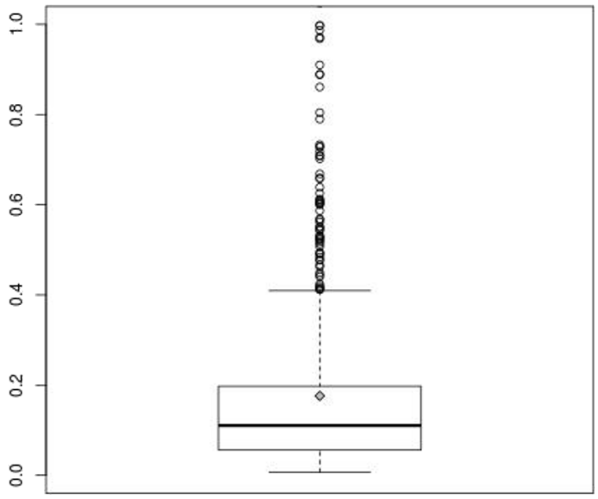


Figure 1. Distribution of productivity of new development projects

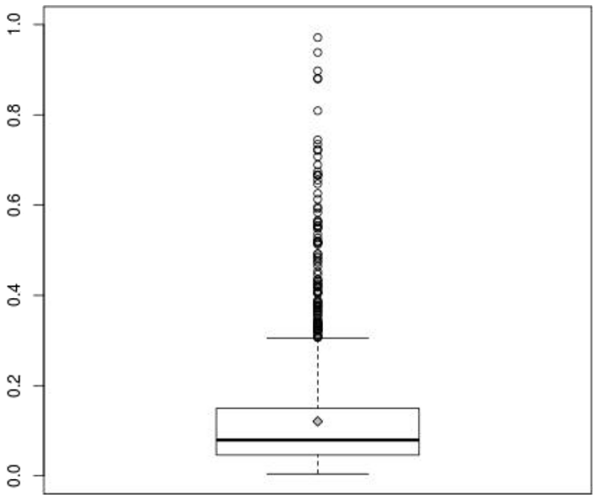


Figure 2. Distribution of productivity of enhancement projects

Figure 2 shows the distribution of the productivity data of the selected projects (the grey diamond is the mean value). Projects with productivity greater than one UFP/PH are not shown, to preserve the readability of the figure.

As the first result of the analysis, one can note that productivity varies widely (the maximum observed value is 2.250% the mean and 66,000% the minimum observed value) and that the productivity of enhancement projects tends to be lower than that of new development projects, but with a smaller variance. This may appear to be somewhat surprising, since the value of UFP for an enhancement project is the size of the part of the application where the enhancement takes place, regardless of the size of the change itself. Therefore, the result shows that, on average, more effort is used in an enhancement project than in a new development project with the same functional size. This is probably due to the fact that maintenance is more challenging than development from scratch.

2.4. Data analysis techniques

We applied several statistical data analysis techniques. The Shapiro–Wilks test was used to check whether specific distributions are normal, and the nonparametric Kruskal–Wallis [13] and Mann–Whitney tests [9] to check if a nominal independent variable affects productivity.

Power law models, i.e. models of the kind $Effort = eUFP^b$, were used to investigate whether a statistical relationship exists between UFP and Effort. Ordinary Least Square (OLS) regression techniques were used after applying logarithmic transformations to both UFP and Effort, because the assumptions about the normality of distributions do not hold for UFP and Effort. Power-law models are used to investigate the existence of economies or diseconomies of scale.

In the paper, the statistical significance threshold is set to 0.05, as customary in Empirical Software Engineering studies. All of the statistical results reported in the paper are statistically significant, i.e. they have p-value < 0.05.

3. Effects of primary programming language on productivity

The impact of the programming language primarily used to develop the project was analysed. Different programming languages call for different development processes, skills, data structures, methods, testing activities, and so on. It is thus reasonable to expect that the productivity of software development may depend on the programming language.

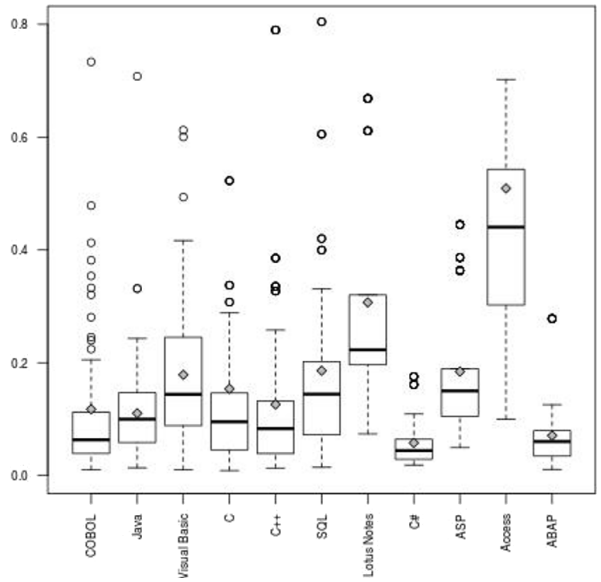


Figure 3. Distributions of new development productivity per programming language

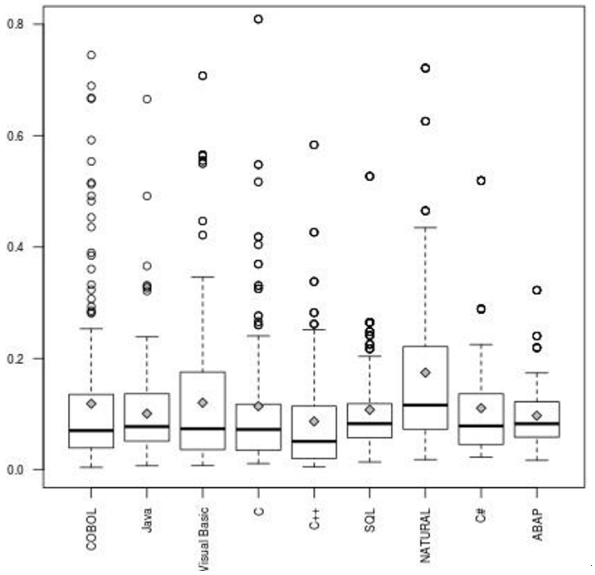


Figure 4. Distributions of enhancement project productivity per programming language

3.1. New development projects

Table 3 gives a few descriptive statistics of new development projects grouped by the programming language. The median productivity greatly changes from a minimum of 0.044 UFP/PH for C# projects to a maximum of 0.425 UFP/PH for access projects. This reinforces the idea that productivity may depend on the programming language.

The distributions of the productivity of projects grouped by language are shown in Figure 3. As the figure shows, the distributions are far from symmetrical, so the "distribution-free" nonparametric Kruskal–Wallis rank sum test [13] was used to assess whether the difference between groups is significant. The results ($\chi^2 = 291.66$, $df = 70$, p-value $< 10^{-15}$) confirm that the primary programming language has a significant effect on productivity.

The authors proceeded to study the effect of the programming languages on productivity for pairs of different programming languages, using the Mann–Whitney test, to check if there was a statistically significant order relationship between the subsets of projects with different pairs of languages. The results are reported in Table 4. The symbol '>' denotes that the projects with the programming language reported in the row of a cell have higher productivity (in a statistically significant sense) than those with the programming reported in the column. Likewise, the symbol '<' denotes the opposite relationship. The symbol '=' denotes that no statistically significant difference was found.

The projects based on the language used in Access appear to be the most productive ones, followed by those based on Lotus Notes. Surprisingly, the productivity of C# projects appears to be the worst one, followed by ABAP, and C++. There is no empirical evidence on the reasons why Access appears very productive while the productivity of C# development appears very low. It can be argued that high-level languages, such as Access, are more productive since they are used in simpler projects and business processes than more complex languages (such as C#) that are generally used in more complex projects of several kinds of application areas.

3.2. Enhancement projects

Table 5 gives a few descriptive statistics for enhancement projects, grouped by their programming language. Note that the languages that appear in Table 5 are not the same as those appearing in Table 3, because in Table 5 the

Table 3. Summary data of new development projects grouped by programming language

Language	N	Median Size [UFP]	Median Effort [PH]	Median [UFP/PH]
COBOL	174	286	4333.5	0.063
Java	114	281.5	3394.5	0.103
Visual Basic	145	327	2760	0.145
C	48	479	4712	0.098
C++	37	312	5100	0.083
SQL	48	615.5	5662.5	0.144
Lotus Notes	16	275.5	1117	0.223
C#	22	285.5	6859.5	0.044
ASP	14	282.5	1957	0.150
Access	24	359.5	845	0.425
ABAP	17	279	6051	0.060

Table 4. Relations between new development productivity of programming languages

Language	COBOL	Java	Visual Basic	C	C++	SQL	Lotus notes	C#	ASP	Access	ABAP
COBOL		<	<	<	=	<	<	>	<	<	=
Java	>		<	=	=	<	<	>	<	<	>
Visual Basic	>	>		>	>	=	<	>	=	<	>
C	>	=	<		=	=	<	>	<	<	>
C++	=	=	<	=		<	<	>	<	<	=
SQL	>	>	=	=	>		<	>	=	<	>
Lotus Notes	>	>	>	>	>	>		>	=	<	>
C#	<	<	<	<	<	<	<		<	<	=
ASP	>	>	=	>	>	=	=	>		<	>
Access	>	>	>	>	>	>	>	>	>		>
ABAP	=	<	<	<	=	<	<	=	<	<	

languages with too few data to support any statistically significant analysis were omitted. The median productivity varies much less than for new development projects, from a minimum of 0.051 UFP/PH for C++ projects to a maximum of 0.116 UFP/PH for NATURAL projects, with the next ones equal to 0.083 UFP/PH for SQL, C#, and ABAP projects. Thus, productivity may depend less on the programming language for enhancement than for new developments.

The distributions of the productivity of projects grouped by programming language are shown in Figure 4.

The comparison of Figures 3 and 4 seems to confirm that the productivity of enhancement projects appears much less dependent on programming languages than the productivity of new development projects. Moreover, it appears that for several languages the productivity in enhancements is substantially lower than the productivity of new developments. The projects based on the NATURAL language [14] are associated with higher productivity than the projects based on other languages. However, in Table 6 the sign '=' occurs more frequently than in Table 4, indicating that the productivities of several language are statistically not discriminated in enhancement projects.

3.3. Comparison with Capers Jones productivity evaluations

Capers Jones [6] studied the relation between the language "level" and its productivity [6]. The language level is defined according to the LoC/FP ratio: the larger the number of lines of code needed to code a Function Point, the lower the level of the language. For example, COBOL requires about 105 statements per FP and is classified as a level 3 language [6]. Table 7 lists the average LoC per FP, the language level, and the average productivity in FP/PM (where PM denotes person-months) according to Jones. In this paper, $PM = PH/160$, where 160 is obtained by multiplying 20 working days per month and 8 working hours per day); the reported data are the result of an analysis concerning software developed up till 1996. To be able to compare our results with those by Capers Jones, the productivity of projects carried out up till 1996 was analysed separately (in columns "Pre" in the tables of this paper) and after 1996 (in columns "Post").

Descriptive statistics are given in Table 8 and Table 9.

These results seem to indicate that there has been a decrease in the productivity for both new developments and enhancements. In the opinion of the authors, the most likely cause is that software complexity has considerably grown, so

Table 5. Summary data of enhancement projects grouped by programming language

Language	N	Median Size [UFP]	Median Effort [PH]	Median Prod. [UFP/PH]
COBOL	306	179	2583	0.070
Java	271	142	2026	0.077
Visual Basic	132	217.5	3154	0.075
C	113	181	2705	0.072
C++	79	141	3810	0.051
SQL	59	142	1837	0.083
NATURAL	55	214	1694	0.116
C#	30	258.5	2728.5	0.083
ABAP	45	249	3069	0.083

Table 6. Relations between enhancement project productivity of programming languages

Language	COBOL	Java	Visual Basic	C	C++	SQL	NATURAL	C#	ABAP
COBOL		=	=	=	>	=	<	=	=
Java	=		=	=	>	=	<	=	=
Visual Basic	=	=		=	>	=	<	=	=
C	=	=	=		=	=	<	=	=
C++	<	<	<	=		<	<	<	<
SQL	=	=	=	=	>		<	=	=
NATURAL	>	>	>	>	>	>		>	>
C#	=	=	=	=	>	=	<		>
ABAP	=	=	=	=	>	=	<	<	

Table 7. Programming language productivity according to Jones (before 1996)

Language	LoC/FP	Level	Avg. Productivity [FP/PM]
ABAP	16	20.0	15 to 30
Access	38	8.5	16 to 23
C	128	2.5	5 to 10
C++	53	6.0	10 to 20
COBOL	107	3.0	5 to 10
DELPHI	29	11.0	16 to 23
Java	53	6.0	10 to 20
SQL	13	25.0	30 to 50
Visual Basic	40	8.0	10 to 20

Table 8. New development projects from the ISBSG dataset: descriptive statistics (Pre: up to 1996, Post: after 1996)

	Size [UFP]		Effort [PH]		Productivity [UFP/PH]	
	Pre	Post	Pre	Post	Pre	Post
Mean	734	583	7319	6607	0.209	0.166
Median	415	303	3703	3074	0.121	0.108
Stdev	822	759	10162	10592	0.342	0.221
Min	53	51	326	320	0.01	0.006
Max	4943	7400	66600	134211	3.96	2.581

Table 9. Enhancement projects from the ISBSG dataset: descriptive statistics (Pre: up till 1996, Post: after 1996)

	Size [UFP]		Effort [PH]		Productivity [UFP/PH]	
Mean	348	290	3750	4098	0.166	0.117
Median	248	161	2104	2193	0.114	0.078
Stdev	376	406	6980	6441	0.155	0.128
Min	52	50	339	322	0.021	0.004
Max	2983	7134	61891	109271	0.939	1.51

Table 10. Comparison with Jones (before 1996)

Language	C. Jones [6] Mean Prod. [FP/PM]	Our analysis Mean Prod. [FP/PM]	Our analysis Stdev/ Mean
C	5 to 10	27	226%
COBOL	5 to 10	23	130%
SQL	30 to 50	33	106%

Table 11. Comparison with Jones (project data up to 2013)

Language	C. Jones [7] Mean Prod. [FP/PM]	Our analysis Mean Prod. [FP/PM]	Our analysis Stdev/ Mean
C	5.62	16.9	99%
COBOL	6.38	15.2	100%
ABAP	7.69	12.4	50%
C++	9.68	13.8	97%
Java	9.68	14.7	66%
C#	9.88	11.7	78%
Visual Basic	13.04	21.8	76%
ASP	13.40	24.1	53%
SQL	15.92	17.3	62%

many technological and methodological advances were "absorbed" by additional difficulty. In fact, the notion of productivity is based on functional size: it is quite possible that modern software has to satisfy more non-functional requirements than old-time software (for instance of security requirements). These additional non-functional requirements certainly require some development effort, which is not explained by the sheer implementation of the required functionality.

In the ISBSG dataset, only three languages were found with enough data points to support a reasonably reliable comparison of productivity before 1996 and after 1996. The comparison – illustrated in Table 10 – is thus limited to these three languages. The columns on the right lists the so-called coefficient of variation, which is the ratio of the standard deviation to the mean, respectively.

Table 10 shows that data from the ISBSG dataset confirm Jones's findings concerning SQL, but indicate that the mean development productivity achieved when using C or COBOL is definitely higher than that found by Jones. It can also be observed that C programming involves a great variability of the productivity level that can be achieved. This is actually not surprising, given that C was used for a wide range of applications and in very different domains.

Table 11 reports an updated set of Jones's productivity data concerning project carried out until 2013 [7].

Table 11 shows that the found mean productivities are greater than those found by Jones. Unfortunately, the authors have no means of explaining this difference. However, there are some similarities between our results and those obtained by Jones: Visual Basic, ASP and SQL appear more productive then the other languages. The main difference is that C appears quite productive according to ISBSG data, while it was ranked as the least productive language by Jones.

It is also possible to observe that the productivity of C programming was less variable after 1996 than earlier. This is probably due to the fact that after 1996 programmers could choose from among so many languages that a relatively low-level language, such as C, is used only in well characterized domains (system-level programming, real-time, etc.).

3.4. Comparison with open-source software development productivity

Delorey et al. analysed 9,999 open-source projects hosted on SourceForge.net to study the productivity of 10 of the most popular programming languages in use in the open-source community [8]. Table 12 reports the data about the languages analysed both in [8] and in this study. The central column in Table 12 provides the data derived from [8] (expressed in Function Points per PH).

With respect to the study by Delorey et al., the data from the ISBSG dataset indicate much higher productivity for all languages. Although this indication is fairly consistent for

Table 12. Comparison with [8]

Language	Delorey et al. [8] Mean Prod. [FP/PH]	Our analysis Mean Prod. [FP/PH]	Our analysis Stdev/Mean
C	0.013	0.120	99%
C#	0.035	0.083	78%
C++	0.032	0.098	97%
Java	0.030	0.105	66%

Table 13. Summary data by business area for new development projects

Business area	N	Median Size [UFP]	Median Effort [PH]	Median Prod. [UFP/PH]
Engineering	18	549.5	1464.5	0.257
Accounting	19	418	4111	0.135
Financial (excl. Banking)	29	327	3123	0.125
Telecommunications	52	262.5	2574.5	0.118
Inventory	12	574	7434.5	0.114
Manufacturing	25	315	3565	0.097
Insurance	38	261	2806.5	0.087
Banking	69	214	2761	0.064

all languages, there is a noticeable difference concerning the C language: while it appears as the least productive language in [8], C appears to be the most productive according to the ISBSG data (in the set of languages considered in Table 12).

4. Effects of business areas on productivity

The previous work [1] reports that the business area can influence development productivity. Accordingly, the dependence of productivity on business areas were analysed here. Projects were thus grouped per business area and only groups of twenty or more projects were kept for statistical analysis.

4.1. New development projects

Table 13 gives the descriptive statistics of new development projects grouped by business areas. The median productivity greatly changes from a minimum of 0.064 UFP/PH for banking projects to a maximum of 0.257 UFP/PH for engineering projects – i.e. the projects supporting various types of activities (design, simulation, etc.) in various engineering areas (civil engineering, electrical engineering, etc.) – approximately four times the minimum. Thus, it can be hypothesized that productivity may depend on the business area. Quite interestingly, the low productivity of insurance projects was already detected in [1] and in [12].

The distributions of the productivity of projects grouped by business area are shown in Figure 5 (where projects with productivity greater than 1 FP/PH are not shown, to preserve the readability of the figure). Since distributions are not symmetrical, the "distribution-free" nonparametric Kruskal–Wallis rank sum test [13] was used to assess whether the difference between groups is significant. The results ($\chi^2 = 116.93$, $df = 76$, p-value $= 0.0018$) confirm that the business area has a significant effect on productivity.

Since the Kruskal–Wallis test only indicates that in at least one case the business area affects the productivity, in this research the Mann–Whitney test was used to study the effect of the business area on productivity for all pairs of different business areas.

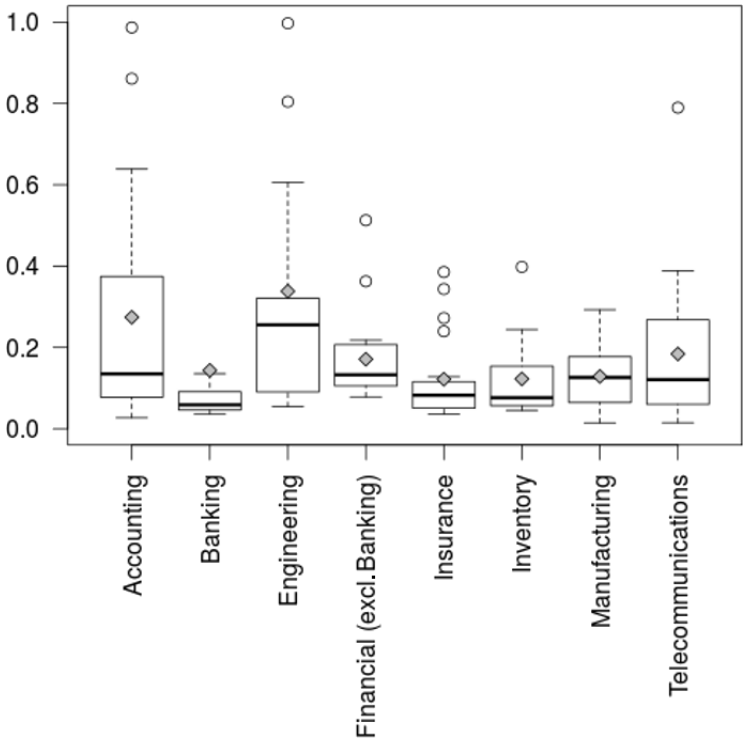


Figure 5. Distributions of new development project productivity per business area

The results of the Mann–Whitney tests are reported in Table 14, with the same conventions as the ones used in Table 6.

The projects belonging to the engineering business area appear to be the most productive ones (as for new developments), followed by those belonging to accounting and financial business areas.

4.2. Enhancement projects

Table 15 gives the descriptive statistics of enhancement projects, grouped by business area. Note that the programming languages that appear in Table 15 are not the same as those appearing in Table 13, because different numbers of data points were available for new developments and enhancement projects and, hence, the areas with too few data to support any statistically significant analysis were excluded from the analysis. The business area with the highest median productivity (Legal – see Fig. 6) has a productivity that is a bit less than five times the lowest median productivity, obtained for Quality. This suggests that productivity may depend on the business area.

The nonparametric Kruskal–Wallis method [13] was used to assess whether the difference between groups was significant. For enhancement projects, the result ($\chi^2 = 119.974$, $df = 44$, p-value $< 10^{-8}$) confirms that the business area has a statistically significant effect on productivity.

Since the Kruskal–Wallis test only indicates that in at least one case the business area affects the productivity, in these investigations the Mann–Whitney test was used to study the effect of the business area on productivity for all pairs of different business areas. The results of the Mann–Whitney tests are reported in Table 16 for enhancement projects.

On the one hand, the legal and insurance projects have the highest enhancement productivity. The insurance projects have high enhancement productivity, while they have quite low development productivity. No data were available to support this kind of analysis, but it can be argued that new insurance projects are less productive since a lot of rules and laws regulate the insurance domain. This requires a lot of effort during the initial phases of the development process, while this effort decreases over time

Table 14. Relations between productivities per business area (new developments)

Business area	Accounting	Banking	Engineering	Financial (excl. Banking)	Insurance	Inventory	Manufacturing	Telecommunications
Accounting		>	=	=	>	=	=	=
Banking	<		<	<	=	=	=	<
Engineering	=	>		=	>	>	>	>
Financial (excl. Banking)	=	>	=		>	=	=	=
Insurance	<	=	<	<		=	=	=
Inventory	=	=	<	=	=		=	=
Manufacturing	=	=	<	=	=	=		=
Telecommunications	=	>	<	=	=	=	=	

Table 15. Summary data by business area for enhancement projects

Business area	N	Median Size [UFP]	Median Effort [PH]	Median Prod. [UFP/PH]
Legal	12	419.5	1485	0.248
Insurance	38	315.5	1679	0.181
Financial (excl. Banking)	44	237.5	1881	0.112
Inbound Logistics	47	106	907	0.093
Outbound Logistics	46	120	1639	0.077
After Sales & Services	26	107	1362.5	0.076
Banking	33	198	2070	0.072
Manufacturing	47	192	3048	0.058
Quality	21	233	3487	0.051
Sales	34	190.5	2609	0.070
Telecommunications	181	142	2151	0.077

whenever legal aspects are well managed. On the other hand, the banking projects confirm their low productivity (for both new developments and enhancements).

5. Effects of architecture on productivity

Different types of architecture call for different development processes, skills and methods. It is thus reasonable to expect that development productivity depends on system architecture. Accordingly, the projects were grouped per architecture and the distributions of productivity were analysed.

5.1. New development projects

The descriptive statistics of the new development project groups characterized by the same architecture are reported in Table 17. Systems with Multi-tier/client-server architecture are characterized by the highest productivity, a bit more than twice the productivity of systems with client-server architecture, the ones with the lowest productivity.

The distributions of the productivity of projects grouped by architecture are shown in Figure 7. The differences in the boxplots do not appear to be large. Since distributions are not symmetrical, the “distribution-free” non-

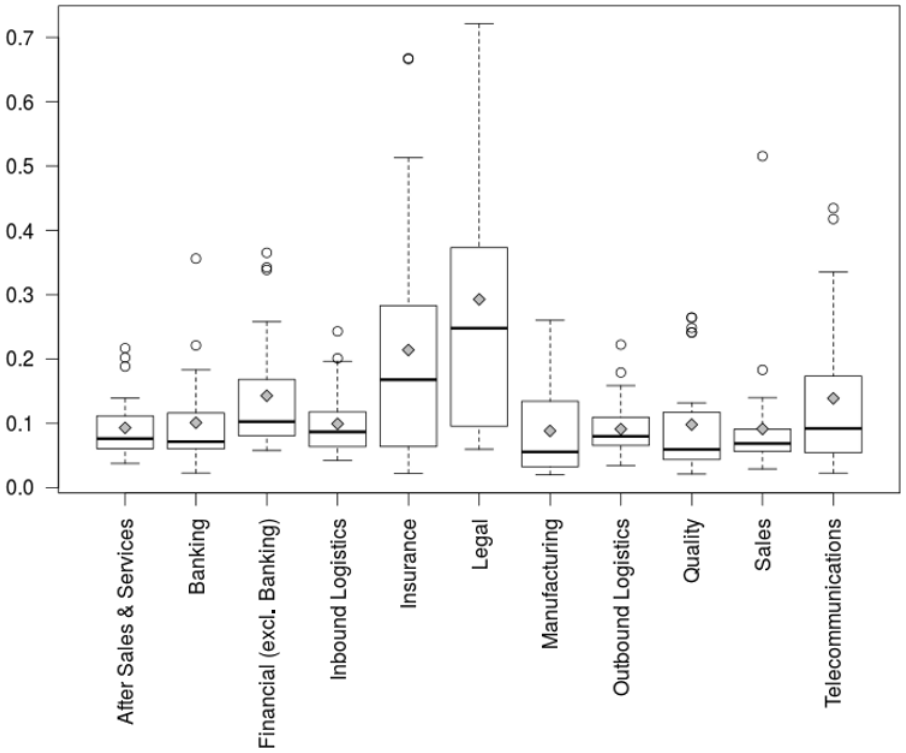


Figure 6. Distributions of enhancement project productivity per business area

Table 16. Relations between business areas (enhancements)

Business area	After Sales & Services	Banking	Financial (excl. Banking)	Inbound Logistics	Insurance	Legal	Manufacturing	Outbound Logistics	Quality	Sales	Telecommunications
After Sales Services		=	<	=	<	<	>	=	=	=	=
Banking	=		<	=	<	<	=	=	=	=	=
Financial (excl. Banking)	>	>		=	<	<	>	>	>	>	>
Inbound Logistics	=	=	=		<	<	>	=	>	>	=
Insurance	>	>	>	>		=	>	>	>	>	>
Legal	>	>	>	>	=		>	>	>	>	>
Manufacturing	<	=	<	<	<	<		<	=	=	=
Outbound Logistics	=	=	<	=	<	<	>		=	=	=
Quality	=	=	<	<	<	<	=	=		=	=
Sales	=	=	<	<	<	<	=	=	=		=
Telecommunications	=	=	<	=	<	<	=	=	=	=	

parametric Kruskal–Wallis rank sum test [13] was used to assess whether the difference between groups is significant. The results ($\chi^2 = 60.45$, $df = 6$, p-value $< 10^{-10}$) confirm that the architecture has a significant effect on productivity.

Here again, the ISBSG dataset does not provide any support for explaining, even tentatively, these results.

Since the Kruskal–Wallis test only indicates that in at least one case the business area affects the productivity, the Mann–Whitney test was used to study the effect of the architecture on productivity for all pairs of different architectures.

The results of the Mann–Whitney tests are reported in Table 18, with the same conventions as the ones used in Table 6.

5.2. Enhancement projects

Table 19 gives a few descriptive statistics of enhancement projects, grouped by architecture. The ratio between the highest and the lowest median productivity is slightly smaller than three. The distributions of the productivity of projects grouped by architecture are shown in Figure 8. The differences in the boxplots do not appear to be large.

The nonparametric Kruskal–Wallis method [13] was used to assess whether the difference between groups was significant. For enhancement projects, the results ($\chi^2 = 45.06$, $df = 6$, p-value $< 10^{-7}$) confirm that the architecture has a significant effect on productivity also for this type of projects. The effect of the architecture on productivity for pairs of different architectures was also studied using the Mann–Whitney test. The results are reported in Table 20.

Multi-tier projects are the least productive for both new development and enhancement projects, while multi-tier with web public interface projects appear to be the most productive just for enhancement projects. Multi-tier/Client server projects are the most productive for new developments, and they maintain high productivity also in the case of enhancement projects.

6. Effects of case tool usage on productivity

The use of CASE tools has long been advocated to improve the productivity of software development processes. While traditionally CASE tools were essentially diagramming/modelling tools, which adopted some sort of a semi-formal design language, such as E/R or Data Flow Diagrams, today the concept embraces all sorts of computer-based tools that are meant to support software development activities. Quite noticeably, some tools are meant to support agile development. For instance, there are tools for writing and managing user stories and tools for writing wire frames and GUI mockups, etc. So, in the ISBSG dataset, "CASE" equates to any computer-based tool supporting software development. However, it can be expected that, in most cases represented in the ISBSG dataset, the used CASE tools are traditional.

Although the usage of CASE (Computer-Aided Software Engineering) tools in software development is conceptually a Boolean variable, in the ISBSG dataset there are four possible values: Yes, No, Don't know and Null (i.e. no value was provided). In the analysis of the effects ofCASE tool usage on productivity, the projects for which there is no clear indication of whether CASE tools were used or not were neglected. That is, only the projects having "CASE tool usage" field equal to Yes (497 projects) or No (851 projects) were retained.

As in the previous cases, the nonparametric Kruskal–Wallis [13] was used to assess whether the difference between groups was significant. The results do not support the hypothesis that the usage of CASE tools has a significant effect on productivity for either new development or enhancement projects.

This result is confirmed by the Mann–Whitney tests on pairs.

Table 17. Summary data of new development projects grouped by architecture

Business area	N	Median Size [UFP]	Median Effort [PH]	Median Prod. [UFP/PH]
Multi-tier/Client server	113	410	2519	0.184
Multi-tier with web public interface	46	169	1470	0.140
Stand alone	234	308.5	3047.5	0.114
Multi-tier	24	479	6496	0.094
Client server	223	350	4628	0.079

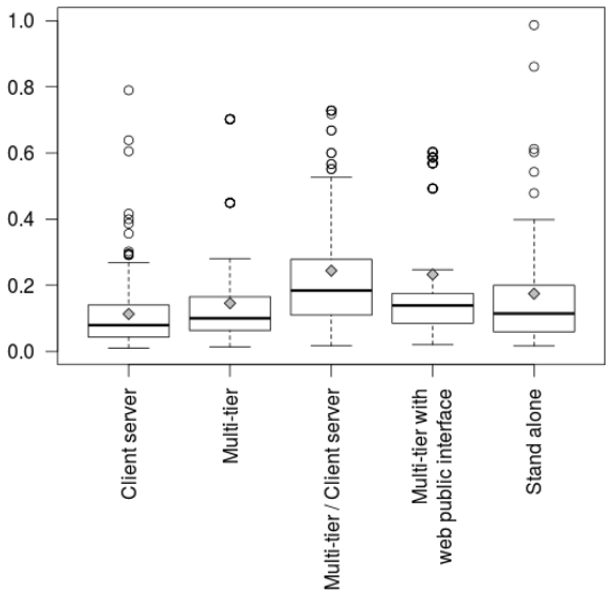


Figure 7. Distributions of productivity per architecture type (new developments)

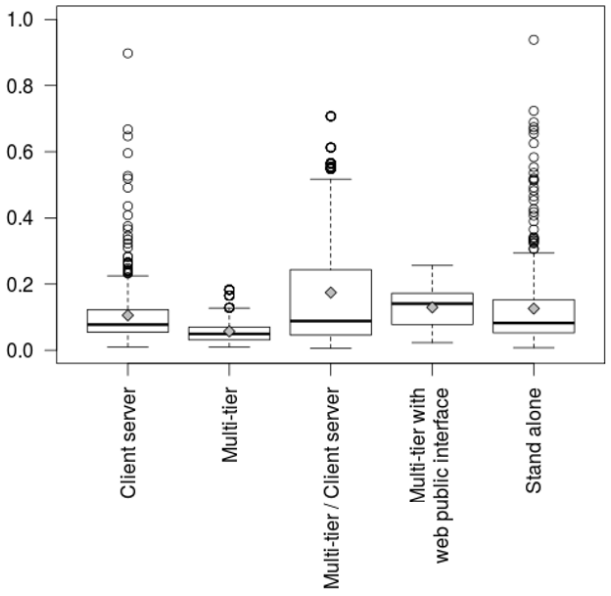


Figure 8. Distributions of enhancement project productivity per architecture type

7. Productivity and economies of scale

The question whether software development exhibits economies (or diseconomies) of scale has been much debated (see Section 9). In general, economies of scale are apparent when it is possible to relate effort and size via models of type $Effort = aSize^b$, with $b < 1$.

In fact, $Effort = aSize^b$ implies that $Productivity = \frac{Size^k}{a}$, where $k = 1 - b$; if $b < 1$, then $k > 0$, and the larger the size, the higher the productivity, as by definition of the economy of scale. On the contrary, if $b > 1$, then $k < 0$, and the larger the size, the smaller the productivity, as in diseconomies of scale. Some studies showed that software development exhibits diseconomies of scale: for instance, this is the case in the well-known COCOMO model [9]. On the contrary, other studies (like [1]) found economies of scale.

To further explore this issue, the existence of $Effort = aSize^b$ models based on ISBSG data was investigated. This type of models is derived by applying the OLS regression after the log-log transformation of data samples. The log-log transformation was used in this research because the data did not comply with the preconditions of OLS about normal distributions.

No statistically significant model could be derived for all new developments, nor for all enhancement projects. Therefore, the economies of scale were studied on data subsets obtained by grouping projects by programming language, business areas, architecture and usage of CASE tools. Grouping project data by these criteria resulted in sufficiently homogeneous datasets, which allowed for the derivation of statistically significant models of effort vs. size.

In the derivation of models, outliers, identified based on Cook's distance, following a consol-

Table 18. Relations between productivities per architecture (new developments)

Architecture	Client server	Multi-tier	Multi-tier/Client server	Multi-tier with web public interf.	Stand alone
Client server		=	<	<	<
Multi-tier	=		<	=	=
Multi-tier/Client server	>	>		>	>
Multi-tier with web public interf.	>	=	<		=
Stand alone	>	=	<	=	

Table 19. Summary data by architecture type for enhancement projects

Architecture	N	Medians Size [UFP]	Effort [PH]	Prod. [UFP/PH]
Client server	443	168	2109	0.078
Multi-tier	45	139	4259	0.049
Multi-tier/Client server	78	339	2860	0.091
Multi-tier with web public interface	51	124	940	0.141
Stand alone	451	175	2096	0.083

idated practice [15] were excluded. The results found are described in the "Outl." column of the tables in the following subsections.

A few statistically significant models featuring quite small adjusted R^2 were found these models are not very interesting, because a small value of R^2 indicates that effort depends mainly on factors other than size and the considered specific characteristics (language, business area, etc.). Accordingly, in the following sections only models featuring adjusted R^2 not less than 0.5 are reported.

7.1. Effect of programming language on economies of scale

By applying the OLS regression after log-log transformation to data samples obtained by grouping new development projects by primary programming language, the models summarized in Table 21 were obtained.

For new development projects that use Java and Visual Basic, the exponent is less than one with 95% confidence: these languages seem to allow for economies of scale. For other languages, it is not possible to decide with 95% confidence if the exponent is less or greater than one, that is, these languages do not cause either economies or diseconomies of scale. It was impossible to obtain statistically significant models only for enhancement projects using PL/1 and ABAP, these are described in Table 22.

PL/I enhancement projects exhibit a diseconomy of scale. Instead, for ABAP enhancement projects no conclusion with 95% confidence could be drawn.

7.2. Effect of business area on economies of scale

By applying OLS regression after log-log transformation to data samples obtained by grouping

Table 20. Relations between productivities per architecture (enhancement projects)

Architecture	Client server	Multi-tier	Multi-tier/Client server	Multi-tier with web public interf.	Stand alone
Client server		$>$	$<$	$<$	$=$
Multi-tier	$<$		$<$	$<$	$<$
Multi-tier/Client server	$>$	$>$		$=$	$=$
Multi-tier with web public interf.	$>$	$>$	$=$		$>$
Stand alone	$=$	$>$	$=$	$<$	

Table 21. Effort models for new development projects grouped by programming languages

Language	Model	Exponent confidence	Adj. R^2	Outl.
C	7.7 $UFP^{1.032}$	0.837–1.226	0.762	3/40
C++	15.6 $UFP^{0.962}$	0.657–1.268	0.622	5/31
Java	37.9 $UFP^{0.769}$	0.656–0.882	0.682	21/107
Oracle	2.8 $UFP^{1.091}$	0.912–1.271	0.852	8/36
SQL	12.0 $UFP^{0.931}$	0.677–1.184	0.549	0/45
Visual Basic	12.8 $UFP^{0.877}$	0.775–0.979	0.714	15/131

Table 22. Effort models for enhancement projects grouped by programming languages

Language	Model	Exponent confidence	Adj. R^2	Outl.
ABAP	7.8 $UFP^{1.069}$	0.909–1.229	0.827	6/45
PL/I	5.7 $UFP^{1.190}$	1.028–1.351	0.658	12/123

Table 23. Effort models for new development projects grouped by business area

Business Area	Model	Exponent confidence	Adj. R^2	Outl.
Financial (no Banking)	8.6 $UFP^{0.955}$	0.672–1.239	0.635	0/28
Telecommunications	12.3 $UFP^{0.915}$	0.675–1.156	0.563	1/47

new development projects per business area, the models summarized in Table 23 were obtained.

The only two statistically significant models found indicate that both economies or diseconomies of scale may occur. The characteristics of effort models for enhancement projects grouped by business area are given in Table 24.

New developments concerning the financial area (excluding banking) appear to allow for economies of scale.

7.3. Effect of architecture on economies of scale

By grouping new development projects per architecture type it was possible to obtain the models summarized in Table 25.

For new development projects, there is no evidence that architectural types lead to economies or diseconomies of scale. By grouping enhancement projects per architecture type, it was possible to obtain the models summarized in Table 26.

Client server and Stand-alone enhancement projects exhibit economies of scale. Although it is not possible to make statements about multi-tier projects with 95% confidence, stil one can observe that the exponent range is mainly less than one in the 95% confidence range, thus it is likely that economies of scale also exist for multi-tier projects.

7.4. Effect of CASE tools on economies of scale

After grouping projects by the usage of CASE tools, the authors were able to find just one model, concerning enhancement projects with the use of CASE tools. The model is described in Table 27.

No economy or diseconomy of scale is apparent.

8. Threats to validity

Construct validity. The definition of productivity is always a sensitive issue and no universally accepted notion of productivity exists. A fairly widely used notion of productivity was chosen for the research, based on the amount of delivered functionality, quantified via UFP, the most widely used functional size measure. Functional size measures, however, may have some weaknesses [16,17], including: (1) the apparent arbitrariness in the selection of the "complexity" weights used to obtain the value of UFP starting from the Base Functional Components (Internal Logical Files, External Interface Files, External Input, External Outputs, and External Queries); (2) the subjectivity inherent to the counting process; (3) the redundancies of the counted elements. As for (1), the weights are based on an initial study by Albrecht [3]. Although they may need to be updated, they are now a part of the standard definition used by ISO for FP [10, 12]. With reference to (2), the International Function Point Users Group periodically issues new guidelines to reduce the amount of uncertainty in the counting process [4]. Finally, the redundancies may affect the efficiency and cost-effectiveness of measuring and using UFP, but are not a real construct threat. However, UFP somehow (and imperfectly) captures the amount of functionality delivered, unlike such measures as LoC which quantify the amount of code delivered and are not available early in the life cycle, but only after coding, when it is too late to make any useful predictions. Also, just because a measure is objectively quantifiable does not mean that it adequately captures a specific software attribute or is useful in practice.

The main threat with this type of studies is the fact that while there are standard definitions of functional size measures, there is hardly any standard definition of how development (or enhancement) effort should be measured. Therefore, different authors may use differently measured effort data. This may lead to different values for productivity.

Therefore, when considering the comparisons reported in Section 4 the reader should take into account the possible differences in effort measures. For instance, the fact that in Table 12 the found productivity values are all greater than those found by Delorey et al. [8] might be due to different effort measurement criteria. In fact, Delorey et al. [8] collected productivity data by

Table 24. Effort models for enhancement projects grouped by business area

Business Area	Model	Exponent confidence	Adj. R^2	Outl.
After Sales & Services	31.3 $UFP^{0.795}$	0.474 –1.116	0.512	1/26
Financial (no Banking)	110.7 $UFP^{0.540}$	0.35–0.729	0.524	13/44
Inbound Logistics	14.5 $UFP^{0.910}$	0.665–1.155	0.574	5/47

Table 25. Effort models for new development projects grouped by architecture

Architecture	Model	Exponent confidence	Adj. R^2	Outl.
Multi-tier	26.1 $UFP^{0.82}$	0.489–1.155	0.548	2/24
Multi-tier Client server	3.5 $UFP^{1.06}$	0.927–1.188	0.746	24/113
Multi-tier with web public interf.	3.2 $UFP^{1.20}$	0.880–1.523	0.626	11/46

Table 26. Effort models for enhancement projects grouped by architecture

Architecture	Model	Exponent confidence	R^2	Outl.
Client server	30.4 $UFP^{0.82}$	0.752–0.898	0.576	77/443
Multi-tier	49.4 $UFP^{0.84}$	0.618–1.062	0.597	5/45
Stand alone	19.3 $UFP^{0.90}$	0.819–0.987	0.539	65/451

analysing the effort devoted by single programmers to single code changes, while the ISBSG collected data concerning whole projects. At any rate, the relative ranking among the various productivities depending on the programming language according to the study of Delorey et al. and according to this study may still be considered valid.

Finally, an intrinsic limit of the analysis is due to the usage of functional size measures to size software. In fact, these measures do not represent the non-functional parts of requirements. So, developing a project with a relatively small functional requirement but huge non-functional requirements (entailing security, reliability, robustness, portability, etc.) may appear unduly characterized by low productivity.

External validity. The obtained results are based on one of the largest datasets publicly available, with projects coming from many different organizations and countries, so they should be fairly representative of the population of new and enhancements projects.

Even though the ISBSG dataset contains a large number of projects, some skew is possible. For instance, some self-selection phenomenon, e.g. only well-organized projects may report their data to the ISBSG dataset, may not be excluded. However, this is a threat that is hard to eliminate for all datasets that collect data on a voluntary basis.

It is true, however, that a large part of the projects in the ISBSG dataset are representative of consolidated practices and languages, instead of innovative ones. The ISBSG dataset does contain data on projects that are recent and innovative, but not enough to allow for a sensible statistical analysis. However, there is a suspicion that innovative applications will always be in the minority in these datasets, given their recentness. It shouldalso be pointed out that a large number of projects are still carried out with consolidated techniques and languages. For instance, in https://www.tiobe.com/tiobe-index/ the top 50 most popular programming languages are listed, and Java, C and C++ are the top 3.

Internal validity. A possible threat to internal validity may come from the fact that these results are based on projects in which data are collected and later reported to ISBSG. This may not be the case for all projects, but this is a threat for all studies of this kind. To mitigate the possible threat due to the way data are collected and reported to ISBSG, only data of the best two categories were used in the research. Moreover, standard data analysis techniques were used. The

Table 27. Effort models for enhancement projects grouped by the usage of CASE tools

CASE tools used	Model	Exponent confidence	Adj. R^2	Outl.
Yes	17.0 $UFP^{0.94}$	0.843–1.037	0.605	45/285

use of log-log transformations may be a possible threat, because the Least Square Regression is carried out with a different figure of merit than the one it would have without the log-log transformation. However, this transformation was useful because the original data did not comply with the assumptions of the Least Square Regression. Also, log-log transformations are quite common in the Empirical Software Engineering, and specifically in the study of Effort models.

9. Related work

A substantial amount of work was carried out to study the main factors affecting software productivity by proposing and analysing processes, methods, tools, and best practices [18–21]. To the best of the knowledge of the authors, there are three literature reviews on productivity factors in software engineering available in the literature [18, 21, 22]. These works focus on the main dimensions of the product, personnel, project, and process. Each of these dimensions is then characterized by sub-factors: product is related to a specific characterization of software, such as domain, requirements, architecture, code, documentation, interface, size, etc. Personnel factors involve team member capabilities, experience, and motivation. Project factors encompass management aspects, resource constraints, schedule, team communication, staff turnover, etc. Process factors include software methods, tools, customer participation, software lifecycle, and reuse. In this paper, the authors do not focus on a specific dimension, but span their empirical study on the main factors reported in the ISBSG dataset (i.e. primary programming language used to develop each software project, the business area addressed by the project, the architectural type adopted by the project and the use of CASE tools).

Directly referring to the factors analysed in this paper, several studies addresses the relation between programming languages and productivity. For example, in [6,8,23–25] different programming languages are studied to investigate their relation with different code aspects such as program length, programming effort, run-time efficiency, memory consumption, and reliability. In [26], the authors explain productivity in the banking, insurance, manufacturing, wholesale/retail, and public administration sectors, limiting their statistical analysis to 206 business software projects from 26 Finnish companies. In [27], software productivity is studied with a dataset on Chinese software companies. Two research question in this study specifically focus on how the business areas and the primary programming language impact productivity, respectively. As for business areas, low productivity is associated to Telecom and Finance areas, while high productivity is associated to Public Administration, Manufacturing and Energy. In this study, financial projects have high productivity, while manufacturing ones have low productivity. In any case, these results cannot be compared with their outputs since in this study two different datasets were analysed (both for the releases and for geographical locations of the projects). As for the programming language, in [27] it is reported that high level programming languages are found to be more productive (the most productive are ASP, C# and Visual Basic, with a median productivity of 34.68, 18.68, and 9.94 size/effort, respectively).

There has also been a considerable debate regarding economies and diseconomies of scale in software development [9, 28–34]. These studies highlighted that it is quite difficult to determine which factors contribute to producing an overall economy or diseconomy of scale; in fact, different dataset provided different indications. Comstock et al. analysed the ISBSG dataset to derive a model that includes both economies and

diseconomies of scale, and can help managers maximize productivity by determining the optimal project size within a particular environment [35]. They considered the same factors as the ones considered in this paper, but with a few important differences: programming languages were considered only in terms of "3rd generation", "4th generation" and "application generators"; moreover, the team size was included in the independent variables of the effort estimation models. This makes the interpretation of the results provided in [35] somewhat problematic as far as (dis)economies of scales are concerned, Productivity is seen there as dependent on size but also on team size, which in it turn is likely to be determined by the size of the program to be developed. As the authors of that work state, "the very presence of Team Size represents a diseconomy of scale: AFP (the size in Function points) relates to the achievement; Team Size relates to the resources consumed" [35]. In fact, the authors conclude that "development exhibits a strong economy of scale with respect to project size, and a similar diseconomy of scale with respect to team size" [35]. This type of finding is consistent with the goals of Comstock et al., but it is of little help for the goals of this study. So, based on the assumption that the team size is chosen to maximize productivity, or to satisfy possible local needs and constraints, the team size is excluded from the independent variables of effort models. In this way, the model of type $Effort = aSize^b$ is obtained for every factor, thus highlighting the role of the considered factor in determining (dis)economies of scale.

10. Conclusion and future work

Software development productivity is an important subject that has often proven to be quite complex to understand and analyse. This paper highlights a few statistically significant results. These results can be considered reliable, since they are based on the analysis of a large public data repository, which is generally considered to be representative of software development practices.

Specifically, it was found out that the primary programming language had a significant effect on productivity of new development projects. On the contrary, the productivity of enhancement projects appears much less dependent on programming languages. The business area and the architecture have a significant effect on productivity of both new development and enhancement projects. No evidence of the impact of the use of CASE tools on productivity was found, for either new developments or enhancement projects.

In addition, it was found that the productivity of new development projects tends to be higher than that of enhancement projects. Also, the results of our analyses show productivity values obtained that are higher, for each programming language, than those of the reference works on the subject, carried out by Jones, and for open-source software, as reported by Delorey et al.

It was also analysed what factors seem to have an impact on the presence of economies and diseconomies of scale. For instance, economies of scale for new development projects using Java or Visual Basic were found and also diseconomies of scale for enhancement projects concerning applications written in PL/1, while neither economies or diseconomies of scale could be found for other projects. Economies of scale were also found for enhancement projects in the financial area (excluding banking), and for enhancement projects concerning application featuring stand alone or client server architectures.

Future work will focus on:

- investigating whether other factors may influence productivity and the existence of economies or diseconomies of scale;
- carrying out analysis on additional datasets;
- using different measures of productivity, for instance, based on different functional size measures.

Acknowledgments

This work has been partially supported by the "Fondo di ricerca d'Ateneo" funded by the Università degli Studi dell'Insubria.

References

[1] R. Premraj, M. Shepperd, B. Kitchenham, and P. Forselius, "An empirical analysis of software productivity over time," in *Proceedings of the 11th IEEE International Software Metrics Symposium*, ser. METRICS '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 37–37.

[2] ISBSG, "International Software Benchmarking Standards Group – Worldwide software development: The benchmark, release 12," 2015.

[3] A. Albrecht, "Measuring application development productivity," in *Joint SHARE/ GUIDE/IBM Application Development Symposium*. IBM, 1979.

[4] *Function Point Counting Practices Manual – Release 4.2*, International Function Point Users Group, 2004

[5] L. Lavazza, S. Morasca, and D. Tosi, "An empirical study on the effect of programming languages on productivity," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, ser. SAC '16. New York, NY, USA: ACM, 2016, pp. 1434–1439.

[6] C. Jones, *Programming Languages Table. Release 8.2*, Software Productivity Research, Inc., 1996. [Online]. https://engenhariasoftware.files.wordpress.com/2008/06/conversao.pdf

[7] C. Jones, "Function points as a universal software metric," *SIGSOFT Software Engineering Notes*, Vol. 38, No. 4, 2013, pp. 1–27.

[8] D.P. Delorey, C.D. Knutson, and S. Chun, "Do programming languages affect productivity? A case study using data from open source projects," in *Proceedings of the First International Workshop on Emerging Trends in FLOSS Research and Development*, ser. FLOSS '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 8–8.

[9] B.W. Boehm, *Software Engineering Economics*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1981.

[10] *Software Engineering NESMA Functional Size Measurement Method, Version 2.1, Definitions and counting guidelines for the application of Function Point Analysis, International Organization for Standardization*, ISO Std. ISO/IEC 24 750:2005, 2005.

[11] "The performance of real-time, business application and component software projects," The Common Software Measurement International Consortium & The International Software Benchmarking Standards Group, Tech. Rep., 2011.

[12] *Software engineering – IFPUG 4.1. Unadjusted functional size measurement method – Counting Practices Manual*, ISO Std. ISO/IEC 20 926:2003, 2003.

[13] W.H. Kruskal and W.A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, Vol. 47, No. 260, 1952, pp. 583–621. [Online]. http://www.jstor.org/stable/2280779

[14] B.A. Myers, J.F. Pane, and A. Ko, "Natural programming languages and environments," *Commun. ACM*, Vol. 47, No. 9, 2004, pp. 47–52.

[15] L. Lavazza and S. Morasca, "Software effort estimation with a generalized robust linear regression technique," in *16th International Conference on Evaluation Assessment in Software Engineering (EASE 2012)*, 2012, pp. 206–215.

[16] B. Kitchenham, "The problem with function points," *IEEE Software*, Vol. 14, No. 2, 1997, pp. 29–31.

[17] B. Kitchenham, S.L. Pfleeger, and N. Fenton, "Towards a framework for software measurement validation," *IEEE Transactions on Software Engineering*, Vol. 21, No. 12, 1995, pp. 929–944.

[18] B.W. Boehm, "Improving software productivity," *Computer*, Vol. 20, No. 9, 1987, pp. 43–57.

[19] A. Trendowicz and J. Munch, "Factors influencing software development productivity – state-of-the-art and industrial experiences," *Advances in Computers*, Vol. 77, 2009, pp. 185–241.

[20] J. Vosburgh, B. Curtis, R. Wolverton, B. Albert, H. Malec, S. Hoben, and Y. Liu, "Productivity factors and programming environments," in *Proceedings of the 7th International Conference on Software Engineering*, ser. ICSE '84. Piscataway, NJ, USA: IEEE Press, 1984, pp. 143–152. [Online]. http://dl.acm.org/citation.cfm?id=800054.801963

[21] K.D. Maxwell, L. Van Wassenhove, and S. Dutta, "Software development productivity of european space, military, and industrial applications," *IEEE Transactions on Software Engineering*, Vol. 22, No. 10, 1996, pp. 706–718.

[22] S. Wagner and M. Ruhe, "A structured review of productivity factors in software development," Institut für Informatik, Technische Universität München, techreport TUMI0832, 2008.

[23] L. Prechelt, "An empirical comparison of seven programming languages," *Computer*, Vol. 33, No. 10, 2000, pp. 23–29.

[24] K. Kennedy, C. Koelbel, and R. Schreiber, "Defining and measuring the productivity of programming languages," *The International Jour-*

nal of High Performance Computing Applications, Vol. 18, No. 4, 2004, pp. 441–448.

[25] R. Klepper and D. Bock, "Third and fourth generation language productivity differences," *Communications of the ACM*, Vol. 38, No. 9, 1995, pp. 69–79.

[26] K.D. Maxwell and P. Forselius, "Benchmarking software-development productivity," *IEEE Software*, Vol. 17, No. 1, 2000, pp. 80–88.

[27] M. He, M. Li, Q. Wang, Y. Yang, and K. Ye, "An investigation of software development productivity in China," in *International Conference on Software Process*. Springer, 2008, pp. 381–394.

[28] D.L. Nazareth and M.A. Rothenberger, "Assessing the cost-effectiveness of software reuse: A model for planned reuse," *Journal of Systems and Software*, Vol. 73, No. 2, 2004, pp. 245–255.

[29] R.D. Banker and C.F. Kemerer, "Scale economies in new software development," *IEEE Transactions on Software Engineering*, Vol. 15, No. 10, 1989, pp. 1199–1205.

[30] J.E. Matson, B.E. Barrett, and J.M. Mellichamp, "Software development cost estimation using function points," *IEEE Transactions on Software Engineering*, Vol. 20, No. 4, 1994, pp. 275–287.

[31] B.A. Kitchenham, "The question of scale economies in software-why cannot researchers agree?" *Information and Software Technology*, Vol. 44, No. 1, 2002, pp. 13–24.

[32] R.D. Banker, H. Chang, and C.F. Kemerer, "Evidence on economies of scale in software development," *Information and Software Technology*, Vol. 36, No. 5, 1994, pp. 275–282.

[33] B. Kitchenham and E. Mendes, "Software productivity measurement using multiple size measures," *IEEE Transactions on Software Engineering*, Vol. 30, No. 12, 2004, pp. 1023–1035.

[34] F.P. Brooks, Jr. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.

[35] C. Comstock, Z. Jiang, and J. Davies, "Economies and diseconomies of scale in software development," *Journal of Software Maintenance and Evolution*, Vol. 23, No. 8, 2011, pp. 533–548.

e-Informatica Software Engineering Journal, Volume 12, Issue 1, 2018, pages: 51–78, DOI 10.5277/e-Inf180103

Knowledge Management in Software Testing: A Systematic Snowball Literature Review

Krzysztof Wnuk*, Thrinay Garrepalli*

*Software Engineering Research Group, Department of Software Engineering, Blekinge Institute of Technology, Karlskrona, Sweden

krw@bth.se, thga14@student.bth.se

Abstract

Background: Software testing benefits from the usage of Knowledge Management (KM) methods and principles. Thus, there is a need to adopt KM to the software testing core processes and attain the benefits that it provides in terms of cost, quality, etc. Aim: To investigate the usage and implementation of KM for software testing. The major objectives include 1. To identify various software testing aspects that receive more attention while applying KM. 2. To analyse multiple software testing techniques, i.e. test design, test execution and test result analysis and highlight KM involvement in these. 3. To gather challenges faced by industry due to the lack of KM initiatives in software testing.

Method: A systematic literature review (SLR) was conducted utilizing the guidelines for snowballing reviews by Wohlin. The identified studies were analysed in relation to their rigor and relevance to assess the quality of the results.

Results: The initial resulting set provided 4832 studies. From these, 35 peer-reviewed papers were chosen among which 31 are primary, and 4 are secondary studies. The literature review results indicated nine testing aspects being in focus when applying KM within various adaptation contexts and some benefits from KM application. Several challenges were identified, e.g., improper selection and application of better-suited techniques, a low reuse rate of software testing knowledge, barriers in software testing knowledge transfer, no possibility to quickly achieve the most optimum distribution of human resources during testing, etc.

Conclusions: The study brings supporting evidence that the application of KM in software testing is necessary, e.g., to increase test effectiveness, select and apply testing techniques. The study outlines the testing aspects and testing techniques that benefit their users.

Keywords: KM, software testing, knowledge, systematic literature review

1. Introduction

Software testing is a complex task and requires various activities, techniques, tools, and resources [1]. Knowledge Management (KM) is extensively used in software testing and influences software testing processes, methods and models [1]. KM helps to capture, share, distribute, and understand knowledge that helps to create a competitive advantage for organizations [2], e.g., by utilizing previous project experience or sharing testing experience between team members [3–6].

The increasing complexity of software systems combined with the advent of distributed development models put more pressure on software organizations to manage organizational knowledge and intellectual capital. Also, there is a significant loss of intellectual capital due to staff turnover, restricted or limited knowledge [6–8]. The adoption of KM principles can help software testing experts to advance knowledge reuse and to encourage management discus-

sions across the organization. There are numerous benefits of applying KM in software testing such as [3, 4, 8–11]:

- increasing test effectiveness,
- decreasing costs, time and effort,
- determination and application of more suited testing techniques,
- determination and application of more suited testing techniques,
- enhancing the quality of results,
- supporting decision-making process.

Explicit knowledge testing can be documented and accessed by multiple individuals, e.g., in test manuals, procedures, test artifacts, test planning, test design specifications, testing logs [12, 13]. Tacit testing knowledge is subjective and hard to document [12] as it mainly forms test execution experiences and discussions with software testers etc. [14]. Insufficient KM during software testing leads to several negative consequences, e.g., low reuse of software testing knowledge, barriers in software testing knowledge adaption, a poor sharing environment of software testing knowledge, difficulties in optimal planning resources [1, 4].

This study focuses on testing aspects as activities during the testing process and the resulting artefacts, i.e. test planning, execution and test result analysis [5–7,15,16], test case design [9,17,18] and testing phases [14, 19]. It also focuses on testing techniques used to execute a software system and find errors [20]. The aim is to focus on the importance of KM in various software testing aspects as the literature lacks studies which focus on identifying the testing techniques that benefit from KM application. Therefore, this work concentrates on identifying the test design, execution, and analysis techniques that help from the KM application. It also explores the related challenges resulting from insufficient KM.

The paper is organized as follows: Section 2 focuses on giving the necessary theoretical background about KM and software testing and their corresponding practices along with its potential contribution to this study. Section 3 provides the research design details and objectives of this study and the addressed research questions. Section 4 contains the details of the research methodology, including considered methodologies and the conducted data analysis. Section 5 depicts the process of conducting the snowballing iterations while Section 6 analyses the results of the literature review. Section 7 lists the identified challenges and implications for research and practice, while Section 8 discusses the limitations of the study. The conclusions are formed in Section 9.

2. Background and related work

2.1. KM in software testing

Testing experience, as well as testing knowledge, are needed to gain a deeper understanding of the used testing techniques [21, 22]. However, testers do not tend to share the knowledge or information that they gain when using various testing techniques [7]. This implies that they miss an opportunity of sharing experiences and learning from each other, which limits their overall knowledge.

Many testers are self-educated and have limited education on the subject [23]. They require additional training [24]. This limited knowledge also results in a limited view about software testing techniques [25]. Technology transfer between research and industry is often limited, in consequence, not all new testing techniques are directly applied in industry [26].

Testers gain various types of knowledge and experiences from their work in software projects. Sharing this knowledge can help to avoid making similar mistakes and optimize testing activities. Efficient organizational knowledge sharing requires establishing efficient KM practices for knowledge creation, documentation, and management.

The primary objective of KM in software testing is to transfer testing knowledge and experience between individuals in the same way as testing documentation as well as utilizing tacit knowledge for supporting test design, execution, and interpretation. KM supports test planning, test result analysis and test outcomes [27]. The test design phase is also heavily dependent on

KM as it involves findings the test conditions and objectives and choosing the relevant information to implement planned test cases. Knowledge also helps to establish the satisfaction criteria against the testing outcomes.

KM supports testing techniques selection as it is often based on testers' experience and intuition, gained from various sources, such as testing the previous versions of the system, involvement in analysing and fixing the defects, working on development and maintenance as well as working with similar software systems [27]. Finally, KM strategies help to increase the effectiveness and efficiency of product testing [28]. Applying KM in software testing is essential to increase the testing level and enhance software quality [26].

2.2. Related work

Several studies looked into the state of the art solutions and practice of utilizing KM for software testing, e.g., [26]. Desai et al. [6] outlined the challenges faced due to the lack of KM, such as less re-use of software testing knowledge, barriers in the transfer of software testing knowledge, difficulties in achieving the most optimum distribution of human resources, etc. Taipale et al. [29] discussed KM practices in software testing and how to enhance the testing practices using KM strategies in organizational units. Wei et al. [14] discussed the implementation of the KM framework in mobile software systems testing and how it benefits the organization concerning decreased costs and increased productivity. Beer et al. [27] stressed that exploratory testing (described as simultaneous learning, test design, and test execution) requires substantial experience. De Souza et al. [1] discussed KM about software testing aspects, testing processes, test phases, test cases and testing techniques, etc. In a similar way, aspects that are related to KM practices are discusse, they encompass, e.g., KM model, knowledge capturing, knowledge elicitation, knowledge retrieval, knowledge dissemination. KM has been investigated for two decades and many tools and techniques were suggested, e.g., methods, tools, techniques, knowledge ontologies, knowledge maps, intranets, just to name a few. Most of the studies focus on storing explicit rather than tacit knowledge and only some studies provide empirical evidence [4, 6, 7, 29, 30], e.g., storage and re-use of test cases [1]. At the same time, many studies focus on implementing a KM framework to strengthen software testing process [5, 7]. From the surveyed papers, the following research gaps were identified:

– storing tacit knowledge and using appropriate testing aspects and techniques,
– focusing on the testing aspects and testing techniques and their importance in utilizing KM practices.

To summarize, so far no study has focused on identifying what type of knowledge is required to perform a particular kind of software testing techniques. This paper fills this research gap by explicitly focusing on finding out the testing techniques and the testing aspects that benefit from KM.

3. Research questions

This study has two goals: 1) to investigate which software testing aspects and techniques receive more attention when applying KM and 2) to identify the challenges faced due to the lack of KM practices.

These goals are detailed into the three research questions:

– RQ1: What are the KM and testing aspects that receive more attention while applying KM in software testing literature?
  Motivation: RQ1 is inspired by De Souza et al. [1] who conducted a systematic mapping to find out the studies related to KM in software testing. De Souza stated various testing aspects that get attention while applying KM in software testing literature but lacked the analysis of the importance of each testing aspect for KM. This paper focuses on identifying which testing aspects investigated in the literature in empirical studies.
– RQ2: What software testing techniques benefit most from the application of KM practices?

Motivation: RQ2 is partly based on the work of de Souza et al. [1] and Beer and Ramler [27], who claimed that exploratory and Ad-hoc testing techniques benefit from the application of KM. The paper further explores De Souza's findings as well explores more techniques which might be considered as important in the context of KM.

– RQ3: What are the challenges faced due to the lack of KM practices in software testing?
Motivation: RQ3 is inspired by Liu et al. [30] who identified the challenges that are faced due to the lack of KM. This article further explores their findings and identifies additional challenges that are faced due to the lack of KM.

4. Research design and methodology

Many authors stressed the importance of utilizing systematic approaches for building knowledge through literature, such as evidence-based software engineering [31], information systems research [32] and results from synthesis [33]. A systematic literature review study was performed for the needs of this article in which the snowballing literature review method suggested by Wohlin [34] was used, rather than a database search based review because 1) it was difficult to formulate a precise search creating the risk of receiving many irrelevant and superfluous papers [34–36], 2) the interdisciplinary nature of the studied area makes the database selection and the search string construction challenging [34,37], 3) snowballing is comparable to the multiple database searches and 4) it is suitable for expanding existing literature reviews with new aspects.

The principle benefits of utilizing snowballing are that it focuses on the cited or referenced papers, which in comparison with the database approach reduces the noise. Moreover, it is usually true that new studies cite one article among the previous pertinent studies or a systematic literature review study already done in a specific area [34].

Snowballing involves deriving the tentative start set of papers and conducting forward and backward snowballing in iterations. Wohlin proposed to use Google Scholar to discover the start set of papers and to evade the publisher bias [37]. However, in certain circumstances, Google Scholar provides significant noise and low certainty in terms of academic quality [38]. Thereby, the Engineering Village database was selected as the start set identification. Knisley recommended the Engineering Village as a prior database to search for papers in comparison with other databases [38]. Also, it was discovered that the Engineering Village offers auto stemming and related papers availability as additional features.

4.1. Data analysis

The qualitative data collected during the literature review were analysed using the narrative analysis technique that helps to create the narrative summary of the resulting studies for synthesis purposes [39]. The narrative analysis does not focus on one specific theme and therefore helps to discover recurring themes from the obtained data. The narrative analysis was used to develop the paper categorization presented in Section 6.1 and the testing aspect and techniques listed in Sections 6.4 and 6.5 The first and the second authors iteratively analysed the results and developed the themes.

The authors also applied grounded theory analysis [40, 41] mainly because they had pre-considered thought regarding the information they needed, contrary to what is recommended by Glaser and Strauss [42]. In the same vein, thematic analysis was excluded as an alternative analysis approach because it searches for the repetitions of themes within the accessible information [43].

4.2. Snowballing procedure

4.2.1. Deriving the tentative start set of paper

Step 1: Search string and database selection. Getting a representative and precise start set of papers is equally challenging for snowball as it is for the database searches [35]. A compre-

hensive search string was developed avoid the problem of inconsistent terminology.

The search string was formulated based on the research questions and the keywords derived from them, including the synonyms and alternatives. It was iteratively developed and it constantly enhanced available knowledge when relevant papers identified manually were read. When there was agreement and confidence that the search string covered the aspects that were the goal of the study, a pilot search was performed in which the Engineering Village database was queried and the first 500 results were analysed. Both authors screened these results independently and later compared and discussed relevance. The resulting search string terms are outlined in Table 1 and grouped into the two categories connected with the Boolean operators.

Table 1. The keywords used to query the Engineering Village database and identify the start set papers

Software testing keywords
Software testing – A1
Software test – A2
KM related keywords
KM – B1
Tacit knowledge – B2
Explicit knowledge – B3
Knowledge creation – B4
Knowledge acquisition – B5
Knowledge sharing – B6
Knowledge retention – B7
Knowledge valuation – B8
Knowledge use – B9
Knowledge discovery – B10
Knowledge Integration – B11
Knowledge theory – B12
Knowledge – B13
Knowledge engineering – B14
Experience transfer – B15
Technology transfer – B16

The search string run in the Engineering Village database was composed of the following Boolean formula: ("A1" OR "A2") AND ("B1" OR "B2" OR "B3" OR "B4" OR "B5" OR "B6" OR "B7" OR "B8" OR "B9" OR "B10" OR "B11" OR "B12" OR "B13" OR "B14" OR "B15" OR "B16").

Step 2: Tentative start set of papers. The search string was executed in the Engineering Village database and resulted in 4832 hits. Next, the inclusion criteria outlined below were applied, including only the papers written in English (IC1), which resulted in 2774 candidates and additional 85 were removed as they were not peer-reviewed (IC2). Next, the 2689 candidates were screened and 2404 were excluded based on title screening (IC4). The abstracts for the remaining 285 candidates were read and 63 papers were accepted. Later the introduction and conclusion sections of the 63 papers were read and as a result, 32 candidates were kept. Finally, the full papers were read and independent judgments regarding if they should be included or not were performed. The application of all inclusion criteria and the full read resulted in 16 candidate papers. These were analysed looking at their authors and publication venues. There were 3 papers which were excluded because they had a low number of references or citations and were less relevant for the scope of this study. As a result, the 13 papers that were left were heavily cited and had the most relevant references that increased the likelihood of better coverage of relevant studies [34]. The following inclusion criteria were used:

- IC1: Articles that are written in English and are published between 2003–2015. The primary reason behind choosing papers from 2003 or later is that KM initiatives in software testing were established around 2003 [1],
- IC2: Peer-reviewed articles published in relevant venues (conferences, workshops or journals in software engineering, software testing and knowledge management, computer science, information technology and science, computing and computer applications)
- IC3: Articles available in full text
- IC4: Articles that focus on KM practices used for supporting software testing (design, execution, and analysis) and/or deal with the industrial challenges due to the lack of KM under software testing.

4.2.2. Forward and backward snowballing in iterations

On the start set of 13 papers [1, 3–7, 9, 14, 27–30, 44], five iterations of backward and forward snowballing were performed, see Table 2 for details. Backward snowballing was conducted by looking at the references of each paper in parallel with forward snowballing by looking at citations. Google scholar was used to extract the citations for each of the papers. Both references and citations were inserted in an Excel file where both titles and abstracts were collected. The second author screened these citations and references in each of the iterations and categorized them into NO, MAYBE and YES categories. Next, the first author screened the MAYBE and YES papers and used his judgment whether they were relevant. After a discussion and reaching an agreement, the relevant candidates were included in the next iteration. The same inclusion criteria were used for all snowballing iterations.

4.3. Data extraction and synthesis

The data extraction properties outlined in Table 2 were derived during several discussions between the authors. The data were extracted into a spreadsheet where categories are mapped to the research questions. The data analysis checklist was also developed where the fulfillment of each of the aspects could be partial or full.

The second author performed the data extraction, supported by the discussion with the first author. The extracted data were synthesized by performing a narrative analysis as per the guidelines provided by Cruzes et al. [39] and Rodgers et al. [45]. Patterns in data were identified, and these patterns were grouped into various themes. To strengthen reliability, rigor and relevance criteria were applied for each paper, see Section 4.4.

4.4. Quality assessment based on rigor and relevance

The rigor and relevance assessment method was utilized according to the guidelines provided by Ivarsson and Gorschek [46]. Previous authors [47, 48] demonstrated that rubrics built the unwavering quality of the assessments as per the terms of inter-rater agreement among the researchers. The second author performed the data extraction supported by the first author who evaluated the results with objectivity in mind. Each paper was allotted with a score utilizing the objective criteria, customized for this study. No significant changes to the rigor and relevance scores suggested by Ivarsson and Gorschek were made, see Table A in Appendix A.

The secondary studies (literature reviews) were evaluated using different criteria. Firstly, it was evaluated if the motivation behind conducting the literature review was clearly stated. Secondly, the review process was examined, and a search for the precise descriptions of the search strategies and search strings, clear definition of acceptance criteria and unambiguous judgments of the validity of the identified studies was conducted. There was also a search for methodological flaws [49]. Finally, the empirical support for the claims provided by the secondary papers was sought and it was checked how well the empirical data were analysed. The fulfillment of each of the criteria was estimated as *Yes, No, Maybe.*

5. Results of the snowballing iterations

As a result of the above examination 13 papers (marked as P1 [5], P2 [3], P3 [4], P4 [6], P5 [14], P6 [1], P7 [29], P8 [9], P9 [30], P10 [27], P11 [28], P12 [7], P13 [44]) were chosen for the start-set from the 4832 candidate papers obtained from the Engineering Village database. Table 3 presents the summary of the snowballing iterations regarding the number of references and citations screened in each iteration.

Based on backward snowballing in five iterations, 843 references were thoroughly examined and evaluated among which 137 were removed based on the publication type, 7 did not match the Language criteria, 40 were duplicates, 323 were dismissed based on title screening, 84 were dismissed based on the year of publication, 202 were dismissed after reading the abstract, 11

Table 2. Data extraction strategy

Category	Data properties	Mapping to research questions
General information	Author(s), Title, Publication Year, Abstract, Conclusions	RQ1, RQ2, RQ3
Type of Study	Evaluation study, Validation study, Proposing a solution, Opinion papers, Personal experience papers, Observational research	RQ1, RQ2, RQ3
Research methods	Case study, Survey, Mapping study, Experiment, Grounded theory, Action research, Unclear	RQ1, RQ2, RQ3
Study aims research outcomes	Does the study specify aspects of software testing that receive more attention while KM is applied?	RQ1
	Does the study specify any software testing techniques i.e., test design, test execution, test result analysis that benefit from the application of KM in Software testing?	RQ2
	Does the study provide any problems or challenges reported due to lack of KM practices in software testing?	RQ3
Data analysis	Aspects of software testing that receive more attention while KM is applied in software testing are properly specified (yes/no/partially)	RQ1
Data analysis	Software testing techniques that benefit from the application of KM are properly specified (yes/no/partially)	RQ2
Data analysis	Problems or challenges faced due to lack of KM practices in software testing explained (yes/no/partially)	RQ3

were excluded after reading the full text and 26 were dismissed as their full text was not available. Finally, 12 papers were obtained based on backward snowballing in five iterations.

During forward snowballing, 614 citations were analysed in five iterations among which 43 turned out to be duplicates, 89 citations were removed based on the publication type, 248 were excluded based on the title, 203 were removed after reading the abstract, 7 were omitted based on the language in they were published, i.e. other than English, 13 papers were removed after reading the full text and 2 were removed due to the unavailability of a full text. Finally, 10 papers were selected.

6. Literature review results analysis

35 papers were identified in five snowballing iterations among which 31 were primary and 4 were secondary studies. Figure 3 depicts the paper distribution over the years. Only five papers were written between 2003 and 2005 indicating that the research in KM in software testing became more common after 2003. Much of the work under KM in software testing was done during 2006–2009 meaning that the organizations started taking interest in utilizing KM in software testing to gain benefits and overcome the issues associated with software testing due to the lack of KM. Still, we see no clear increasing trend in Figure 1.

Out of 35 analysed papers 21 studies are conference articles indicating that conferences are the primary venue for communicating research in KM for software testing. Journals correspond to 34% of the studies (14 out of 35). Table A in Appendix A provides the list of publication venues. It appears to be clear that not only software engineering venues are utilized for communicating research about KM in software testing.

Next, it was analysed which of the three RQs each of the papers addressed. It turned out that 23 out of 35 studies reported various KM aspects (RQ1) during software testing, 12 papers discussed challenges (RQ3) faced due to the lack of KM practices in software testing, while ten stud-

Table 3. The summary of the number of citations and references screened in each snowballing iteration. I – Iteration, FS: Forward Snowballing, BS: Backward Snowballing, D – Duplicate, T – Based on Type, N – Based on Name, Y – Based on Year, L – Based on Language, EA – Excluded after reading the abstract, EF – Excluded after reading the full text, FN – Full text not available, IA – Included after reading Abstract, IF – Included after reading full text

Iteration	FS/BS	Papers rejected from FS and why	Papers rejected from BS and why	Papers considered from FS and why	Papers considered from BS and why
I1	140/346	D: 6, T: 19, N: 41, EA: 66, L: 3	T: 51, L: 5, D: 16, N: 105, Y: 44, EA: 87, EF: 11, FN: 20	IA: 4, IF: 2	IA: 4, IF: 2
I2	164/262	D: 13, T: 31, N: 55, L: 4, EA: 54, EF: 4	N: 135, D: 5, Y: 21, T: 37, L: 2, FN: 5, EA: 53	IA: 2, IF: 1	IF: 4
I3	294/178	D: 19, T: 39, N: 145, FN: 2, EA: 79, EF: 9	T: 44, D: 13, N: 61, Y: 14, EA: 44, FN: 1	IF: 1	IA: 1
I4	12/50	D: 4, N: 5, EA: 3	D: 4, N: 19, T: 5, Y: 5, EA: 16	–	IF: 1
I5	4/7	D: 1, N: 2, EA: 1	D: 2, N: 3, EA: 2	–	–

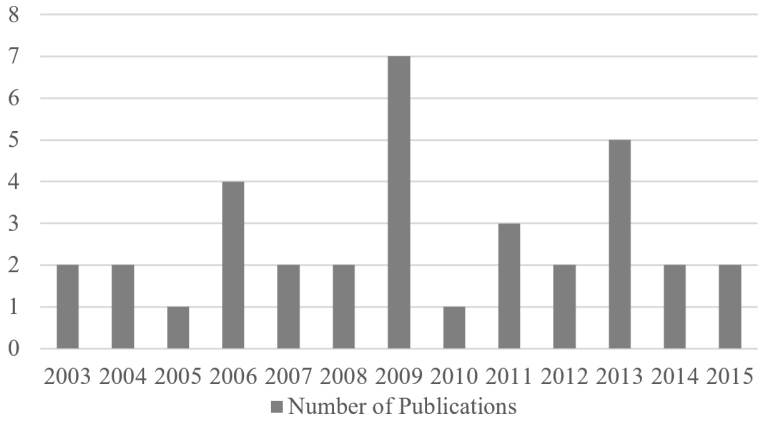


Figure 1. Publications over the years

ies focused on KM in testing techniques (RQ2) and helping testers to select better testing techniques.

6.1. Categorization of papers based on research methodology and studytype

In the analysed group 31 studies were primary studies and four were secondary studies (two systematic literature reviews and two systematic mapping studies). The 31 primary studies were categorized according to the research methodology (i.e. case study, survey, experiment, etc. as defined by Runeson et al. [50] and the type of study (i.e. evaluation, proposal, solution, opinion, experience based, etc. and constraints as defined by Wieringa et al. [51].

Evaluation research which utilized the case study research method dominated among the chosen papers – 16 articles [P2, P3, P4, P5, P8, P11, P12, P13, P18, P20, P25, P30, P31, P33, P34, P35] of which 3 were interview studies [P2, P11, P31], categorized as qualitative case studies. Evaluations using frameworks were found in 5 papers [P9, P14, P15, P27, P29]. The framework-proposal category encompassed 4 papers [P17, P22, P23, P26]. Two papers

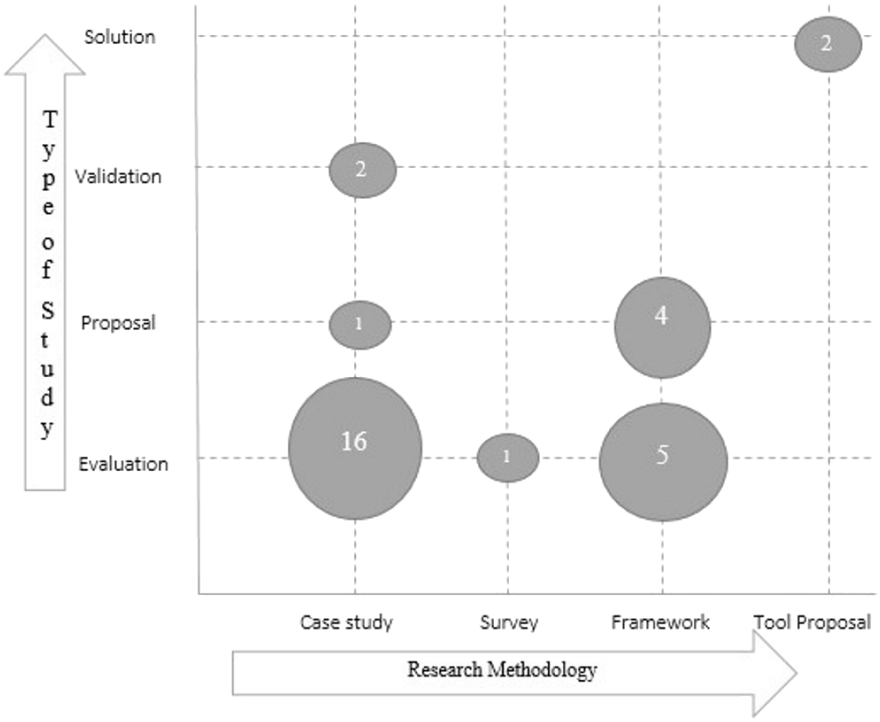


Figure 2. Categorization based on the type of the study and the research methodology aspects

were classified as case study-validation [P7, P10] and two papers as the tool a proposal-solution [P21, P32]. Finally, the categories such as case study-proposal [P19] and survey-evaluation [P1] received only one paper, see Figure 2 for details.

6.2. Quality assessment based on rigor and relevance

Figure 3 depicts the Rigor and Relevance analysis results where the primary studies are categorized into four quadrants (A, B, C and D) according to their rigor and relevance scores. The process of classification is detailed below.

- Papers which fall under the score from (0–1.5) are categorized as low rigor and those that fall in between the score of 2 as high rigor.
- Papers with the score from (0–2) are considered to have low relevance and the papers that fall score 2.5 or above are considered to have high relevance.

Altogether 13 studies were characterised as having high rigor and high relevance, quadrant A in Figure 3, and these outcomes are the most reliable. Also, 12 studies were classified under quadrant C with high relevance and low rigor. Six papers fell under category D, which means they were characterised by low rigor and low relevance, where relevance scores prevail over rigor scores, see Table B in Appendix B for rigor and relevance scores.

6.2.1. Quality assessment of secondary studies

Table 4 shows the results of the quality assessment of secondary studies [P6, P16, P24, P28]. It was concluded that the four identified secondary studies present high quality and therefore trustable literature reviews.

6.3. KM aspects discussed in the selected studies (RQ1)

The subject of 23 studies were KM aspects which testers focus on during software testing, see Table 5. It occurred that 13 studies focused on knowledge representation while 12 studies focused on knowledge capturing. There were 8 papers which focused on knowledge management systems and 8 papers presented knowledge management models.

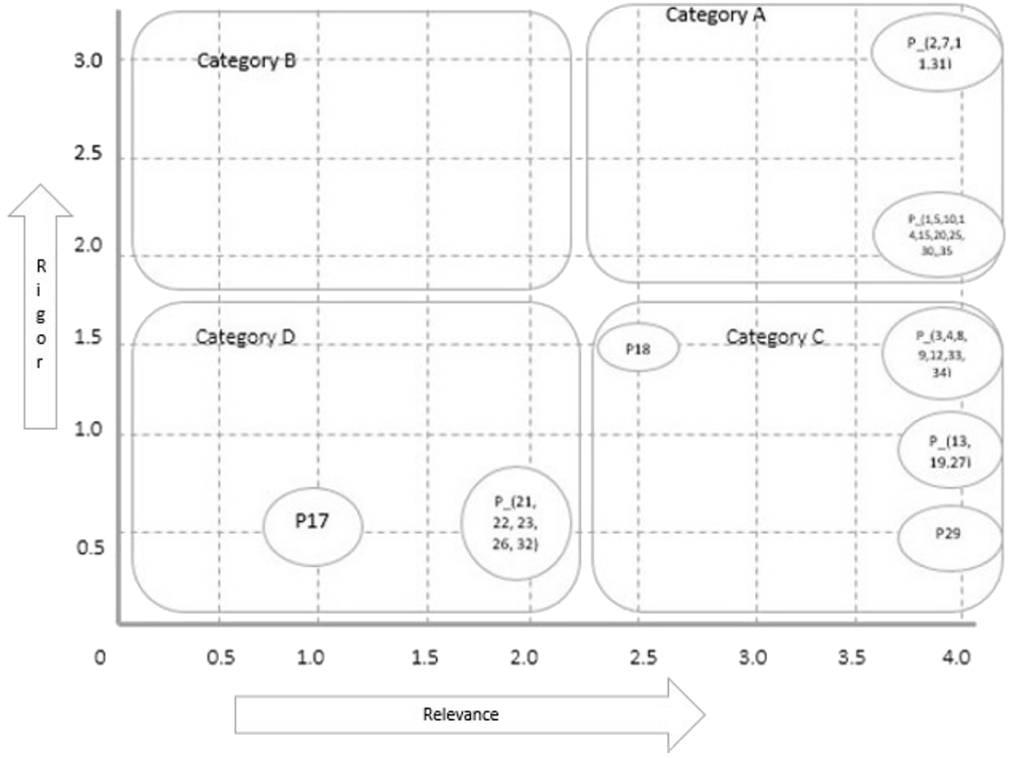


Figure 3. Rigor and relevance analysis results

Table 4. Quality assessment for secondary studies

Quality assessment question	P6	P16	P24	P28
Is the motivation behind conducting systematic literature review and mapping clearly expressed and defined?	Yes	Yes	Yes	Yes
Is the process of conducting systematic literature review or mapping clearly stated?	Yes	Yes	Yes	Partial
Is there any empirical evidence for the stated systematic literature review or mapping study?	Yes	Yes	Yes	Yes

KM systems (KMS) are necessary to enable successful KM. Huseman and Goodman [52] consider KMS as an essential source for competitive advantage while Rajiv and Sarvary [53] claim that organizations without strong KM systems work inefficiently, which consequently influences their quality of work.

Eight studies [P1*, P3, P4, P5*, P8, P9, P12, P19] (in this notation an asterisk (*) indicates a paper with high rigor and relevance scores) proposed various KMSs and discussed their importance for software testing. KMS were used to store, manage, search and share various kinds of knowledge with the help of knowledge documents [P3, P4, P9, P19], to store tacit knowledge to be reused by searching the relevant documents and resolve any raising issues [P12] or store and maintain daily and weekly tester discussions in a knowledge map [P8] or, also, store the experience gained in earlier testing cycles [P5*].

KMS provide several benefits, e.g., they help to reduce effort during testing, increase software quality [P1*, P5*], help the organizations adapt to turnover and faster respond to changes and downsize by making an experience of each individual widely accessible [P4]. It is interesting to note that all the eight papers focused on the importance of KMS and their benefits, rather than the details of how these systems are built and what strategies were used during their development. Thus, future research should focus on the strategies to be used to

Table 5. Research Focus on KM aspects over the years

KM aspect	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
KM System (KMS)	–	–	–	P12	P5*	P19	P3 P9	–	P1* P4	P8	–	–	–
KM Models	P27	–	–	–	–	P19	P3 P9	–	P1*	P8 P25*	P2*	–	–
Knowledge Representation	–	P15*	P14*	P18 P29	P5*	P19	P3 P9	–	P34	P8	P32 P33	P22	–
Knowledge capturing	–	P15*	P14*	P18	P5*	P19	P3 P9 P35*	P26 P35*	P1*	–	P2*	P21	–
Knowledge retrieval	–	P15*	P14*	P18	P5*	P19	P3 P9	–	–	–	P2*	–	–
Knowledge dissemination	–	–	–	P23	–	–	–	–	P1*	–	P2*	P21	–
Knowledge elicitation	P27	–	–	–	P5*	–	–	–	–	–	P2*	–	–
Knowledge packaging	–	P15*	P14*	P18	–	–	–	–	P1*	–	–	–	–
Knowledge evolution	–	P15*	P14*	P18	–	–	–	–	–	P8	–	–	–
Knowledge acquisition	–	–	–	P29	–	–	–	–	–	–	P32 P2*	P21	–
General[a]	P17	P28	–	–	P6 P7*	P10*	P13 P20* P30*	–	–	–	P16 P11* P24	P6	P31*

[a]Not focusing on any KM aspects but provides tools that support KM and knowledge or just defining KM aspects without implement them.

build an effective KMS and its usage in case study context.

KM models are models used for knowledge management and knowledge process aspects, such as knowledge carriers and knowledge technologies. Eight studies [P1*, P2*, P3, P8, P9, P19, P25*, P27] focused on KM models. Three studies [P3, P9, P19] used communication databases enriched by knowledge maps and testing knowledge databases. KM models can also be created based on reusable test case repositories extracted from similar projects or individuals' tacit knowledge and testing projects data by test specialists [P8].

KM models bring several benefits, e.g., increase test case reuse [P8], increase quality and decrease development time [P1*], develop testing lesson learned systems [P2*], or identify gaps in KM practices and fill in these gaps with potential solutions [P27].

Two models for building KM models were identified. The first model contains four phases: 1) absorption is related to acquiring new knowledge from the external environment of the organization, i.e., experts are brought into the organization, 2) diffusion concerns the dissemination of knowledge among individuals in the organization, i.e., these issues which are mostly resolved in email/discussion lists, search engines, best practices, 3) generation involves the improvement of new knowledge and the procedure of turning tacit knowledge into explicit information, i.e. through brainstorming sessions, joint design and source studies, 4) exploitation is referred to as the commercialization of knowledge [P27]. The second model contains five steps: 1) identify knowledge needs, 2) create knowledge, 3) store knowledge, 4) organize knowledge and 5) share knowledge [P25*].

The identified KM models focus on acquiring, improving, disseminating and storing testing knowledge. These findings may help to understand that KM models contributed to the increase in the reuse of testing knowledge in some papers [P25*, P27, P3, P9, P18].

Knowledge representation focuses on representing test knowledge through various tools that support knowledge storage, e.g., ontologies, Software Requirement Specifications (SRSs), Test Procedure Specifications (TPSs), etc. Thirteen studies [P3, P5*, P8, P9, P14*, P15*, P18, P19, P22, P29, P32, P33, P34] focused on knowledge representations which were categorized into:

- Ontologies [P3, P8, P9, P19, P22, P29, P32], TPSs and SRSs as explicit knowledge representations. Ontologies serve as a medium in describing relative concepts, attributes and relations connected with knowledge [P3, P9, P19], they are also used to generate test cases for GUI testing [P34], or as the knowledge representation for performance testing [P22]. Ontologies were also used as knowledge representations for test case reuse [P8] and for supporting acquisition, organization, reuse and sharing testing knowledge [P29, P32]. Testing activities can be performed based on the ontologies associated with a software project [P29, P32]. Ontologies support test case generation from various artefacts in dissimilar domains [P33] or for organizational discussions [P2]. These results suggest that developing an ontology that possesses all of the above characteristics could result in generating productive testing outcomes. It is also worth exploring how to use these ontologies and strategies rather than how to develop them [P3, P9, P19, P22, P29, P32].
- Characterization schema [P14*] that contains test objectives, test scope, required testing technique, test case generations, and test tools is applied in post-project evaluations and summaries of experiences from testing activities. A characterization schema is a tool that supports knowledge representation. Vegas et al. developed and empirically evaluated the schema for assisting testing technique selection that generates a valid test case for a given project [P14*]. This study suggests effective schema generation for test design technique selection.

Knowledge capturing includes codifying and documenting analytical testing knowledge in a manner that individuals can adapt and re-use for specific purposes. 13 studies [P1*, P2*, P3, P5*, P9, P14*, P15*, P18, P19, P21, P26, P33, P35*] focus on capturing testing knowledge in terms of using: 1) lessons learned, experiences, successes and failures [P2*], 2) knowledge of individuals from discussion forums and documents [P3, P9, P19], 3) external knowledge and its relation to internal knowledge [P1*], 4) feedback given by both producers and consumers using characterization schemata [P14*, P15*, P18], 5) experience and knowledge gained from applying various testing techniques [P26]. Three papers specified capturing general testing knowledge, e.g., knowledge and experience are recorded and represented to as a substantial quantity of component sequence in an XML file [P35*], recorded into a formal form (issue spreadsheet) [P5*] or in wikis [P21].

What is surprising is that the identified studies focus on Externalization (tacit to explicit), Internalization (explicit to tacit) aspects leaving aside Socialization (tacit to tacit) and combination (grouping all the explicit knowledge).

Knowledge acquisition is the focus of four studies [P2, P21, P29, P32] with the help of wikis [P21], ontologies [P29, P32] or lessons learned [P2]. Surprisingly, the studies do not outline any process that needs to be executed while defining the rules unlike [P14*] which outlined such a 10-step process for knowledge capture. It can thus be concluded that researchers should focus on knowledge acquisition processes and techniques.

Knowledge elicitation is the focus of three papers [P2*, P5*, P27]. They utilized: 1) an architectural model for knowledge elicitation based on the lesson learned systems (a KM manager as well as expert testers verify the elicited knowledge) [P2*], 2) eliciting expert knowledge whenever it is required and capturing it in spreadsheets [P5*] or 3) acquiring knowledge from the external environment during the absorption

phase [P27]. The architectural model presented by Andrade et al. [P2*] focuses on 1) defining the structure of software testing lessons learned, 2) setting up the procedures for the management of lessons learned and 3) supporting the design of tools that manage lessons learned. Despite promising results, papers [P5*, P27] proposed only the knowledge elicitation tools and failed to provide the processes for knowledge elicitation.

Knowledge dissemination covers disseminating testing knowledge through various KM practices, such as internalization, externalization, combination, and socialization. Only four studies [P1*, P2*, P21, P23] focused on the ways to disseminate testing knowledge. In two studies [P1*, P21], knowledge is available in a useful, readable format to the individuals who need it. Andrade et al. used active knowledge dissemination, where the software testing lesson learned systems disseminate the lessons learned as per various parameters (e.g., scattering of conceivably helpful lessons learned towards the beginning of every testing activity using a testing activity descriptor). The second way is passive knowledge dissemination where the user is responsible for communicating the software testing lessons learned system and asking for the conveyance of lessons learned [P2*]. Lee developed a KM framework with seven cyclic steps for disseminating testing knowledge: identify relevant knowledge, collect the knowledge that is needed, adapt knowledge, organize knowledge in a readable format and apply the knowledge assets to situations where there is a need for it [P23].

Knowledge retrieval covers returning testing information in a structured format contrary to just capturing the knowledge. Eight studies [P2*, P3, P5*, P9, P14*, P15*, P18, P19] focused on knowledge retrieval mechanisms and tools or artefacts that support them. Three studies [P14*, P15*, P18] provided a systematic structured format of storing the knowledge regardless of the testing technique.

Knowledge packing covers strategies or methods used in packing captured knowledge, e.g., knowledge databases. In [P14*, P15*, P18], a characterization schema encompassing various attribute levels, such as tactical, operational and historical, was developed for packaging the experience of individuals for various testing activities. In [P1], knowledge packing is done with the aid of a KM System by following the knowledge lifecycle from acquisition to an application.

Knowledge evolution covers evolution aspects, such as the evaluation and maintenance of testing knowledge. There were four studies [P8, P14, P15, P18] covering this aspect. Three studies propose a characterization schema [P14*, P15*, P18] where a librarian maintains the repository by taking care of the coherence of the information it contains and updates the repository based on the feedback provided by consumers and producers. In one study, a knowledge analyst is assigned to analyse conducted discussions and update the knowledge repository [P8]. Three studies consider knowledge evaluation as the most important element [P14*, P15*, P18] but fail to provide methods, steps and strategies for supporting knowledge evolution and thereby recommendations for software organizations.

6.4. Software testing aspects that benefit from the application of KM practices (RQ1)

In the studied group 9 studies [P2, P3, P6, P9, P16, P19, P23, P24, P27] provide only a general discussion about KM and how to apply knowledge in software testing, however, they lack discussions on specific testing aspects. Two studies [P5, P7] focused on dealing with KM applied in a project where testing is outsourced to a third party (this is not considered a testing aspect). The only difference is that the process is carried out elsewhere but all the activities of this process are similar. The remaining papers are analysed according to the following testing aspects that are summarized in Table 6 and the research focus on the testing aspects over the years is summarized in Table 7.

Testing process. Seven studies [P1*, P4, P12, P21, P25*, P29, P32] focus on KM in the context of a testing process (test planning, test case design, test execution and test result analysis), see Table 7.

Table 6. Description of the testing aspects analysed in the study

Testing aspect	KM utilization
Testing process (7 studies)	**Test planning:** The main aim is to manage knowledge in the context of test scenario creation, test cases design, data preparation as well as a test environment. **Test execution:** the goal is to manage knowledge during test execution, in a number of test cycles on the basis of the project. For example, most of the projects run two test cycles by adhering to time and cost conditions. **Test result analysis:** The aim is to manage knowledge during test result analysis.
Test cases and code (3 studies)	Manage knowledge about the test cases. For example: reusing the test cases Manage knowledge about the test code (which takes into account test scripts and drivers)
Testing phases (2 studies)	Apply KM strategies to phases in software testing such as unit testing, component testing, integration testing, system testing, acceptance testing, alpha testing, beta testing.
Testing techniques (10 studies)	Manage knowledge on testing techniques, to help testers to choose a better suited testing technique for designing test cases, executing and analysing the tests.
Testing types (6 studies)	Manage knowledge in a specific software testing type such as GUI, load testing etc.
Testing resources and tools (2 studies)	Manage knowledge about usage of testing tools or resources.

Table 7. Research focus on testing aspects over the years

KM aspect	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
Testing process	–	–	–	P12 P29	–	–	–	–	P1* P4	P25*	P32	P21	–
Test case and code	–	–	–	–	–	–	P35*	P26	–	P8	–	–	–
Test levels and testing phases	–	–	–	P29	P5*	–	–	–	–	–	–	–	–
Testing technique	P17	P15* P28	P14*	P18	–	P10*	P13 P20* P30*	–	–	–	P11*	–	–
Testing type	–	–	–	P12	–	–	P33 P35*	–	P34	–	–	P22	P31*
Testing resource	–	–	–	P29	–	P10*	–	–	–	–	–	–	–
General	P27	–	–	P23	P5* P7*	P19	P3 P9	–	–	–	P2* P16 P24	P6	–

Abdullah et al. [P1*] focus on utilizing a community of practice for managing testing knowledge. Their model involves community of practice, KMS functionality in software testing and KMS architecture. The software testing process structures the testing knowledge, i.e. it begins with the system requirements for product specification and development, which comprises system design and coding, and proceeds to the verification and validation of product.

Desai et al. [P4] reported that KMS integrated with software testing, data warehousing, and mining helps to store and retrieve relevant knowledge and to discover different modules which are scattered along memory locations. This improves the software testing process, including test planning, test case development, test execution, test result analysis and reporting.

Nogeste et al. [P12] concluded that applying KM improves test planning and increases tacit knowledge capturing for subjects not experiences in KM. Abdou et al. [P21] advocate that a software testing process should be enriched with solutions used by Open Source Software communities regarding test planning, forming test design, considering test specifications, test implementation, deriving test cases or test suites, test execution, accepting test results.

Sirathienchai et al. [P25*] proposed three models for test planning, test preparation and test reporting which leverage on KM. Firstly, cost assessment is performed, followed by the performance evaluation of the software testing process performed by different experienced personnel utilizing the project duration, cost, and quality. Finally, a comparative financial analysis is done to find the best solution by return on investment, payback, and benefit cost ratio. The findings from the case study revealed that the long-term continuous investment on KM can improve the testing process performance more efficiently than the short-term counterpart.

Barbosa [P29] defined a testing process based on an ontology that combines the development paradigm and the testing strategy. This ontology (Reference Ontology on Software Testing) captures the relevant testing knowledge and stores it in a repository. Individuals can use the stored knowledge during testing. The main difference between ROost and other ontologies is that ROost was developed mainly following a well-established method named SABiO, which was used in several ontology development efforts [54]. ROost covers aspects related to the software testing process and its activities, artefacts that are utilized and produced by the activities, testing techniques for test case design and test environment including human, software and hardware resources similar to OntoTest [P29] with a prime motive to manage testing knowledge.

Test case and test code. Three studies [P8, P26, P35*] discussed the use of KM in test cases and test code reuse. Li et al. used an ontology representation and a knowledge model [P8] for test case reuse. Upheld by the management level, a testing center built a reusable test case repository with more than 12,000 cases, complemented with an organizational exchange library. The case study results demonstrate that the effectiveness and efficiency of the test case design and the work circumstances of test engineers and managers were improved.

Nasser et.al [P26] suggested a knowledge-based software test generation framework that permits to characterize the domain and system specific coverage criteria for different software artefacts and domains, specifically concentrating on test cases. By utilizing the custom coverage criteria, test specialists can control what tests are to be incorporated in generated test suites. For this reason, the framework used reasoning with ontologies to address the test case selection issue for re-use. Based on the ontology, test individuals can choose and select relevant previous test cases for a given project.

Li [P35*] presented a test case generation model based on test code reuse for GUI testing. The testers experience is recorded and represented as extensive segments in an XML document, where segments are the instances defined in a GUI ontology. Next, all components that are related to data elements are distinguished and marked in a sequence which is connected to data elements. In step 4, for each sequence set, data dependent elements among user related

components are recorded. In step 5, for every sort of knowledge components, the outcomes with comparative sensible relations are figured out, and if the recurrence of a legitimate connection surpasses the normal level, they are concluded as a rule for test case generation. This approach was evaluated in a case study which indicated that test case generation for GUI testing was found more efficient.

Test levels or testing phase. Two studies [P5*, P29] focused on the application of KM at a specific testing level or phases such as unit, integration or system testing. Wei and Ying [P5*] emphasizes that to deliver high quality and high productivity of testing during system testing, a KM framework should be integrated into the organization in such a way that test knowledge can be shared between individuals in the organization to sustain the system test and maintain the quality of testing. Barbosa et al. [P29] suggested an ontology which captures all the relevant knowledge that takes place during the testing phases and stores it in the repository.

Testing techniques. Ten studies [P10*, P11*, P13, P14*, P15*, P17, P18, P20*, P28, P30*] discussed KM and software testing techniques. Test case generation is one of the leading aspects of software testing and is closely linked to the selection of testing techniques [55]. Vegas and Basili [P14*, P15*] proposed a characterization schema that includes comprehensibility, the maturity level of the individuals performing testing, cost of application, inputs, dependencies, repeatability, software type, experience required to use a given technique and knowledge required to apply this technique. Beer and Ramler [P10*] claimed that ad hoc testing, subsuming casual testing methodologies and exploratory testing, is benefited through the application of KM practices, where test case design and execution are interwoven to design new test using the information and experience gained constantly.

Itkonen et al. [P11*] suggested that exploratory testing techniques benefit from adapting KM practices. Knowledge in exploratory testing can be utilized as data to guide exploratory test design and to perceive failures, e.g., as a test oracle to differentiate between an expected correct outcome and an incorrect defective outcome [56]. Moreover, knowledge together with the observed actual behavior of the tested system can be utilized to make new better tests during exploratory testing. The authors also found that as the domain knowledge, the system knowledge and generic knowledge are required to recognize failures.

Koznov et al. [P13] claimed that one of the main obstacles in transferring formal methods to industry is a lack of KM methods in this area and focusing on explicit rather than tacit knowledge, e.g., model based testing needs well defined and documented requirements which are not set in industrial projects. Tinkham and Kaner [P17] listed the factors which contribute to a tester's choice of exploration style, such as tester's skills, experience, detailed knowledge on the usage of a technique and personality (including learning style). All these factors are essentially for a perfect utilization of exploratory testing which happens through capturing, storing testing knowledge which, in turn, can be done through KM practices. Itkonen et al. [P20*] indicated that knowledge engineering techniques play a crucial role for more effective use of testing techniques.

Testing type. It deals with selecting software aspects to be tested, while the testing techniques deal with how a specific part of the software will be tested. Six studies [P12, P22, P31*, P33, P34, P35*] focused on managing knowledge in a specific software testing type, such as performance, GUI, endurance testing.

Nogeste and Walker [P12] conducted a case study which proved that a KM based regression process is necessary since regression testing is heavily dependent on tacit and explicit knowledge identification, collection, sharing and documentation.

Frietas and Vieria [P22] developed an ontology for the core knowledge used for performance testing. Since ontologies serve as the representation of domain knowledge that empowers knowledge sharing among different applications, the paper investigated the impact of ontologies on performance testing. The results indicate that this ontology can also be extended

to endurance and stress testing both of which are subclasses of performance testing for better results.

Valeh et al. [P33] applied knowledge management techniques in automated software testing to enhance the control over test generation. The results indicate that the use of ontology brings benefits for the automated testing specification of extensible test oracles which can model test specialists' mental model and lend themselves to define custom coverage criteria. The system grants control to a test specialist to determine or indicate which test cases ought to be produced and generated to increase the quality of test suites. Moreover, the produced test suite ontology is programming language independent and can be deciphered into various languages and reused.

Gentry and Shirazi [P31*] discovered that Canadian software development organizations utilize in-house manual software testers when tacit knowledge is obliged to successfully test a software application. Software development companies will probably keep manual testing in-house, since the relationships between testers and other internal employees may build the viability of testing. Software development organizations are more averse to outsource manual testing when domain specific knowledge is essential to test the product.

Nasser et al. [P34] proposed ontology-based test case generation to facilitate GUI testing and produce test cases from the users' viewpoint. GUI testing is knowledge-intensive and requires both the knowledge of GUI systems and extensive experience, hence a knowledge-based technique was suggested.

Li et al. [P35*] proposed an ontology based semi-automatic approach to generate test cases using testers' experience. The approach is based on a GUI testing ontology and examines the source code with reverse engineering techniques. Secondly, the test case generation rules are extracted from the testers' experience. The evaluation results indicate that the usage of knowledge representations and management provides support in test case generation for GUI testing in terms of greater efficiency.

Testing resources or tools. They represent resources that can be humans (testers, test managers or test analysts) or hardware (equipment, software, testing tools or supporting systems). Hardware and software resources are characterized as the testing environment which can be utilized to automate the testing methods. Two studies [P10*, P29] discuss KM concerning testing tools and resources. Beer and Ramler [P10*] focus on experience with tools when planning test case automation. Extensive experience with the setup and the utilization of tools was required and indicated as a critical issue for producing reliable test results. Barbosa et al. [P29] classified the software resources needed to perform testing (including testing tools) into primary, organizational and supporting tools.

6.5. Software testing techniques that benefit from the application of KM practices (RQ2)

Model-based testing benefits from the application of KM practices [P13]. Exploratory (ad hoc) testing is mentioned as a testing type in a few papers such as [P17], but it is also called as a testing technique in a few papers such as [P6, P10*, P16] and a testing approach also in a few papers [P11*]. In this study exploratory testing is considered as a testing technique because it is recognized as test design by [57, 58].

7. Challenges due to lack of KM practices

Twelve papers [P1*, P3, P4, P6, P7*, P9, P16, P19, P20*, P22, P25*, P32] discussed challenges faced due to the lack of KM practices, they are outlined in Table 8.

CH1: Low software testing knowledge reuse rate [P1*, P3, P4, P6, P9, P19] due to the lack of KM practices, learning and knowledge reuse are limited. Failure to capture individual knowledge and experience leads to repeating the same mistakes even though there are individuals in the organization rectify mistakes or prevent them from reoccurring. Low testing knowledge reuse

Table 8. Challenges faced due to lack of KM in software testing

Challenges faced	P1*	P3	P4	P6	P7*	P9	P16	P19	P20*	P22	P25*	P32
CH1: Low reuse rate of software testing knowledge	x	x	x	x		x		x				
CH2: Barriers in Software testing knowledge transfer	x	x	x	x		x	x	x				
CH3: Poor knowledge sharing environment	x	x	x	x		x		x				
CH4: A serious loss of software testing knowledge		x	x	x		x		x				
CH5: Optimal distribution of human resources quickly		x	x			x		x				
CH6: Determining if adequate testing has been performed			x									
CH7: Difficulties in achieving test coverage			x									
CH8: Determining if the outputs are correct			x									
CH9: Documentation is not being updated					x							
CH10: Troubleshooting documentation was inaccurate					x							
CH11: Insufficient schedule and release information					x							
CH12: Define satisfaction criteria.			x									
CH13: Increase in cost and time							x					x
CH14: Decreasing test effectiveness												x
CH15: Lacked practices for logging and tracking in testing									x			
CH16: Knowledge exchange							x		x			
CH17: Identify whether the most critical aspects of test components are tested									x			
CH18: Less support for decision making										x		
CH19: Testing knowledge not considered in planing				x								
CH20: Lacking skills											x	
CH21: Missing high severity defects											x	

also increases the effort to accomplish a task in software testing. Even if an organization has a few testing knowledge databases, most of the staff neglect to use them without the aid of KM practices, which contributes to low test knowledge reuse.

CH2: Barriers in software testing knowledge transfer [P1*, P3, P4, P6, P9, P19] and knowledge transfer without the proper application of KM practices are challenging. Also, individuals always search for the knowledge that they require and do not search the entire repository. Yellow pages can serve as a medium for rectifying this problem. Moreover, IT staff is not able to understand new testing knowledge without the aid of KM practices. The reason for this is that most of the knowledge in organizations is tacit, obtained through experience and difficult to articulate. KM representation technologies help to overcome this challenge.

CH3: Poor sharing environment for software testing knowledge [P1*, P3, P4, P6, P9, P19], the lack of a formally established, unique and sorted knowledge sharing environment negatively impacts communication. A knowledge sharing model as indicated by Sirathienchei [16] has to be accumulated within an organization to overcome this issue.

CH4: Serious loss of software testing knowledge [P3, P4, P6, P9, P19], the insufficient application of KM practices leads to knowledge and experience accumulation around only a few members of staff. Therefore, maintaining knowledge repositories and databases that store individual knowledge and make use of it is required. Also, a sudden staff turnover leads to the loss of testing knowledge.

CH5: No possibility to quickly achieve the most optimum distribution of human resources [P3, P4, P6, P9, P19], KM helps to integrate humans, processes, and technology. In a situation when management does not have any idea about the staff's knowledge level, even an ideal team will not be optimally formed in testing projects which have negative impact on achieving the optimum distribution of human resources [4].

CH6: Determination whether adequate testing is done [P4], the application of knowledge as a test oracle gives answers to the question when testing should be stopped and points out whether adequate testing is done or not. Therefore, with the help of KM practices, this issue can be resolved [6].

CH7: Difficulties in achieving test coverage [P4], the lack of KM practices hinders the identification of the untested parts of the code base. Moreover, another challenge is the fact that reusable test cases may be neglected and not stored in the repository, which increases the testing effort.

CH8: Determination whether the outputs are correct or not [P4], knowledge can be used as a test oracle to identify whether the obtained code execution results acomply with the expected outcomes [17]. Thereby, the lack of KM practices may have a negative impact on determining whether the outputs are correct because relevant knowledge is neglected.

CH9: Documentation is not updated [P7*], updating knowledge repositories is rarely done, which results in outdated repositories and relying on them when a problem occurs provides inaccurate results [29]. In such a case, knowledge evolution and maintenance methods help to allocate knowledge analysts or a specially selected person, e.g., a librarian who maintains the repository by taking care of the coherence of the information it contains and updates the repository regularly as indicated by Vegas et al. [59].

CH10: Troubleshooting documentation is inaccurate [P7*], knowledge documents that retrieve human knowledge, such as expert knowledge, are not efficiently maintained [29]. Knowledge managers and experts are to be allocated to check knowledge databases as well as to verify the knowledge that is accumulated and stored in the repository and rectify occurring problems as indicated by Andrade et al. [3].

CH11: Schedule and release of information from the testing organization to development are found to be insufficient [P7*], the documentation was not up-to-date and insufficient for planning.

CH12: Determination what decision should be made about the software when testing is completed, whether to proceed further and develop satisfaction criteria [P4].

CH13: Increase in cost and time [P32] due to the lack of relevant knowledge.
CH14: Decreasing test effectiveness [P32] because essential testing knowledge is not available.
CH15: Less support for decision making [P22] as critical knowledge is not available when needed.
CH16: Testing knowledge not adequately considered for test planning [P6] and test execution.
CH17: Insufficient test technique skills [P25*] since the test team consists of several roles which encompass different responsibilities and knowledge that needs to be communicated and shared.
CH18: No high severity defect detection is another challenge faced due to the lack of KM practices in software testing [P25*].
CH19: No methods for logging in and tracking testing activities based on experience [P20*].
CH20: Transfer of the required knowledge to testers and utilizing it [P20*].
CH21: Focusing testers' attention to ensure that the most important aspects of the tested features are tested [P20*].

7.1. Implications for research and practice

The analysis of the **KM aspects discussed in the selected studies (RQ1)** brings several implications for research and practice. Firstly, there appears to be a lot of focus on knowledge representation and knowledge capturing. This focus is unsurprising as it results in a rather technical focus on KM application, creating or managing knowledge databases or repositories or building additional tools into the testing environment, which allows for the development of enhanced knowledge documentation. Secondly, knowledge acquisition or elicitation received little attention in the surveyed papers. This has implications for software testing, especially for software companies that base their products on the OSS code or other external sources. These companies need to be more active in knowledge acquisition or elicitation since extensive knowledge is available in OSS communities (also testing experience or competence). Thirdly, knowledge dissemination (especially outside the testing teams) received little attention. However, the authors believe this aspect will be dominant in the successful testing of software products that are greatly based on open source software or external sources. For example, efficient testing knowledge dissemination with other companies involved in OSS communities can help to reduce testing costs and efforts as the communities can take over large parts of testing responsibilities. Fourthly, not much e attention was devoted to understanding how testing knowledge was created, especially tacit knowledge. Since many software companies work in Agile-inspired environments, it is believed that focusing on tacit knowledge management remains critical here. Fifthly, most of the papers [P1, P2, P3, P6, P9 and P19] identified or discussed some testing aspects but failed to discuss their importance, or connected these aspects (e.g., test case and testing phase) to testing processes (e.g., test planning, test case design, test execution and test result analysis) [1]. It is postulated that researchers should adjust the focus of research endeavors and introduce some of these aspects into exploring KM for software testing.

Looking at the **importance of additional testing techniques and types (RQ1 and RQ2)**, a possible implication from these results is that the suggested techniques and types are seldom validated. Moreover, it remains greatly unclear which testing methods to use in each of the software testing activities. Therefore, researchers should focus more on creating operational guidelines regarding which testing methods to use for which activities. Next, regression testing and GUI testing are considered to gain strong benefits from using the ontologies or KM models. More research needs to be conducted to provide similar analysis and clearly identify what testing techniques require what type of knowledge and how much these testing techniques are sensitive to, e.g., eliciting or creating tacit knowledge. Most of the studies have not specified and have not focused on the knowledge relevant for a specific testing technique. The taxonomy that summarizes the types of knowledge that support various testing techniques and their types is what is clearly missing in the current literature.
Focusing on tacit knowledge remains important since no study has focused on identifying the

importance of tacit knowledge management for software testing. It is also important to explore the importance of tacit knowledge and how to identify it, capture it and store it. In addition, there is a need to explore testing aspects as well as determine what testing types are dependent on efficient tacit knowledge management.

Identify mitigation strategies for the identified challenges (RQ3), there is a need to identify the mitigation strategies concerning each of the identified challenges and provide tools, recommendations, and techniques to overcome those challenges. The most distinct challenges are associated with knowledge reuse, knowledge transfer or knowledge sharing (CH1, CH2, CH3), and they clearly show that more research focus should be given to these areas. From the point of view of software companies, these areas will become dominant in the next years as more software is co-created in open source software communities or externally acquired from external software organizations. Moreover, insufficient knowledge sharing or transfer often results in losing the knowledge that is critical and therefore substantial additional costs are borne when restoring this knowledge. Thus, it is postulated that researchers in KM and testing should broaden their focus areas and expand the technical aspects by adding human aspects, knowledge reuse topics as well as organizational aspects that lead to increased knowledge sharing.

8. Validity threats

Validity threats under the snowballing phase of the thesis are discussed according to the four validity categories suggested by Wohlin et al. [60].

Internal validity threats are minimized by creating and maintaining a review protocol which encompassed the details of the search string formulation and start set identification, inclusion and exclusion criteria used, the quality assessment being carried out, etc. The risk for judgment error was minimized by performing the independent evaluation of the two authors who later compared and discussed the results. Both authors worked closely together and discussed any questionable cases. Moreover, internal validity threats are mitigated by following the mapping guidelines provided by Petersen et al. [61] and quality assessment criteria as per the guidelines provided by Ivarsson and Gorschek [46]. Finally, there is still some risk that the studied positive testing outcomes are the result of other aspects than applying KM techniques. It is planned to explore this aspect in future work when these relationships are explored in detail.

Construct validity focuses on various potential confounding factors regardless of whether a study could capture the intended knowledge, i.e. to achieve the aims and objectives. One of the main concerns for this research is multiple definitions of KM. This threat was mitigated by adopting the well cited definition by Davenport [2]. As indicated by Kaner [62], construct validity depends on the question of "How does one recognize that they are measuring what they usually think they are measuring against?". The search string structure could be one of the construct validity threats in this study. Therefore, the search string was iteratively formulated with extensive discussions between the authors. Next, data extraction could also be the source of validity threats. To avoid these threats, supervisor's assistance was accepted and all updates at each step were sent for approval.

External validity considers the capability to generalize results outside the studied context. Most of the studies fall under the case study research category with high rigor and relevance scores as most of them were conducted in industrial contexts. Thus, the outcomes can be considered industry pertinent and are more generalized. For the studies that received low rigor and relevance scores, it remains to be determined if the ideas suggested in these studies have high generalizability.

Reliability considers the degree of repeatability and whether the data and analysis depend on a specific researcher. To strengthen reliability, each step of the snowballing process was documented, including the database search. The same applies to each step of data collection and analysis and they can be backtracked, if needed. The quality assessment of the chosen papers was

ensured by using rigor and relevance criteria according to objective assessment criteria. The properties and aspects identified from the papers were mapped with the research questions to achieve the objectives of the study.

9. Conclusions

Software testing is knowledge-intensive and the use of KM practices and tools provides a wide range of benefits regarding the increase in capital and quality [1]. This paper focuses on the implementation of KM in software testing and on exploring the importance of KM in each of the software testing aspects and testing techniques. Also, the paper presents the challenges faced due to the lack of KM in software testing. The topic is explored in a systematic literature review.

Looking at the testing aspects identified in the study (RQ1), the results indicated that KM is mainly used to support the selection or execution of testing techniques (10 studies) or optimization of the testing processes (7 studies). At the same time, managing testing resources or knowledge about test cases or the test code has been greatly underrepresented. Knowledge elicitation, dissemination, acquisition, evolution and packaging receive little attention in the surveyed literature indicating that knowledge is mainly managed during software testing within a project or an organization and less attention is devoted to further knowledge sharing. Knowledge management system, models, representation, capturing and retrieval are the main KM areas that the surveyed literature focuses on.

Looking at the testing techniques that benefit from the application of KM practices (RQ2) the results indicate that ad hoc and exploratory testing gain more benefits from utilizing KM techniques than model-based testing techniques. This appears to be logical since model-based testing operates on highly formalized knowledge (models) where extensive reasoning can frequently be applied. Ad hoc or exploratory techniques rely more heavily on tacit knowledge and therefore demand more KM techniques.

This study identifies 21 challenges faced due to the lack of KM practices in software engineering (RQ3) and the most frequently mentioned challenges are associated with testing knowledge reuse, transfer, and sharing. Moreover, the risk of losing testing knowledge appears to be one of the prominent challenges. To summarize, this paper has made the following contributions:

- Exploring various testing aspects that are focused on while KM is applied in software testing literature. Moreover, the importance of each of the software testing aspect concerning KM was explored.
- Discovering that each of the testing aspects is focused on while KM is applied, albeit few of them are very important in the KM context.
- Determining the importance of each of the software testing techniques (i.e. design, execution and result analysis techniques) in the KM context along with obtaining the knowledge which is required for each technique so as to provide recommendations to store the tacit knowledge just in case any technique turns out to be important in the context of KM and utilize tacit knowledge.
- Uncovering various challenges that are faced due to the lack of KM in software testing literature

In future work, the authors plan to conduct case studies and investigate how KM is utilized during software testing by software-intensive organizations. There are also plans to explore the enabling factors that allow achieving good testing coverage without KM techniques. It is planned to study what modeling framework and models can support software testing tacit knowledge capture, analysis, storing and reuse. Finally, tacit knowledge management in software testing will also become the focus of further studies.

Acknowledgments

This work is supported by the IKNOWDM project from the Knowledge Foundation in Sweden (20150033).

References

[1] E.F. de Souza, R. de Almeida Falbo, and N.L. Vijaykumar, "Knowledge management initiatives in software testing: A mapping study," *Information and Software Technology*, Vol. 57, 2015, pp. 378–391. [Online]. http://www.sciencedirect.com/science/article/pii/S0950584914001335

[2] T.H. Davenport and L. Prusak, *Working knowledge: How organizations manage what they know.* Harvard Business Press, 1998.

[3] J. Andrade, J. Ares, M.A. Martínez, J. Pazos, S. Rodríguez, J. Romera, and S. Suárez, "An architectural model for software testing lesson learned systems," *Information and Software Technology*, Vol. 55, No. 1, 2013, pp. 18–34.

[4] Y. Liu, J. Wu, X. Liu, and G. Gu, "Investigation of knowledge management methods in software testing process," in *International Conference on Information Technology and Computer Science*, Vol. 2. IEEE, 2009, pp. 90–94.

[5] R. Abdullah, Z.D. Eri, and A.M. Talib, "A model of knowledge management system in managing knowledge of software testing environment," in *5th Malaysian Conference in Software Engineering (MySEC)*. IEEE, 2011, pp. 229–233.

[6] A. Desai and S. Shah, "Knowledge management and software testing," in *Proceedings of the International Conference & Workshop on Emerging Trends in Technology*. ACM, 2011, pp. 767–770.

[7] K. Nogeste and D.H. Walker, "Using knowledge management to revise software-testing processes," *Journal of Workplace Learning*, Vol. 18, No. 1, 2006, pp. 6–27.

[8] L. Xu-Xiang and W.N. Zhang, "The PDCA-based software testing improvement framework," in *International Conference on Apperceiving Computing and Intelligence Analysis (ICACIA)*. IEEE, 2010, pp. 490–494.

[9] X. Li and W. Zhang, "Ontology-based testing platform for reusing," in *Sixth International Conference on Internet Computing for Science and Engineering (ICICSE)*. IEEE, 2012, pp. 86–89.

[10] J. Kajihara, G. Amamiya, and T. Saya, "Learning from bugs (software quality control)," *IEEE Software*, Vol. 10, No. 5, 1993, pp. 46–54.

[11] C. O'Dell and C. Jackson Grayson Jr, "Knowledge transfer: discover your value proposition," *Strategy & Leadership*, Vol. 27, No. 2, 1999, pp. 10–15.

[12] I. Nonaka and H. Takeuchi, *The knowledge-creating company: How Japanese companies create the dynamics of innovation.* Oxford University Press, 1995.

[13] A.D. Marwick, "Knowledge management technology," *IBM Systems Journal*, Vol. 40, No. 4, 2001, pp. 814–830.

[14] O.K. Wei and T.M. Ying, "Knowledge management approach in mobile software system testing," in *IEEE International Conference on Industrial Engineering and Engineering Management*. IEEE, 2007, pp. 2120–2123.

[15] T. Abdou and P. Kamthan, "A knowledge management approach for testing open source software systems," in *International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2014, pp. 1–2.

[16] J. Sirathienchai, P. Sophatsathit, and D. Dechawatanapaisal, "Simulation-based evaluation for the impact of personnel capability on software testing performance," *Journal of Software Engineering and Applications*, Vol. 5, No. 08, 2012, p. 545.

[17] V.H. Nasser, W. Du, and D. MacIsaac, "An ontology-based software test generation framework." in *The 22nd International Conference on Software Engineering and Knowledge Engineering, SEKE*, 2010, pp. 192–197.

[18] H. Li, F. Chen, H. Yang, H. Guo, W.C.C. Chu, and Y. Yang, "An ontology-based approach for gui testing," in *33rd Annual IEEE International Computer Software and Applications Conference*, Vol. 1. IEEE, 2009, pp. 632–633.

[19] E.F. Barbosa, E.Y. Nakagawa, and J.C. Maldonado, "Towards the establishment of an ontology of software testing." in *International Conference on Software Engineering & Knowledge Engineering*, 2006, pp. 522–525.

[20] N. Juristo, A.M. Moreno, and S. Vegas, "Reviewing 25 years of testing technique experiments," *Empirical Software Engineering*, Vol. 9, No. 1, 2004, pp. 7–44.

[21] A.C.C. Natali, A.R.C. da Rocha, G.H. Travassos, and P.G. Mian, "Integrating verification and validation techniques knowledge into software engineering environments," *Proceedings of 4as Jornadas Ibeoamericanas de Ingeniería del Software e Ingeniería del Conocimiento, JIISIC*, Vol. 4, 2004, pp. 419–430.

[22] N. Juristo, A.M. Moreno, and S. Vegas, "Towards building a solid empirical body of knowledge in testing techniques," *ACM SIGSOFT Software Engineering Notes*, Vol. 29, No. 5, 2004, pp. 1–4.

[23] R.L. Glass, R. Collard, A. Bertolino, J. Bach, and C. Kaner, "Software testing and industry needs," *IEEE Software*, Vol. 23, No. 4, 2006, pp. 55–57.

[24] R. Jain and S. Richardson, "Knowledge partitioning and knowledge transfer mechanisms in software testing: An empirical investigation," in *Proceedings of the 1st Workshop on Advances and Innovations in Systems Testing*, 2007.

[25] S. Vegas, "Identifying the relevant information for software testing technique selection," in *International Symposium on Empirical Software Engineering, ISESE*. IEEE, 2004, pp. 39–48.

[26] X. Liu, G. Gu, L. Yongpu, and W. Ji, "Research and application of knowledge management model oriented software testing process," in *11th Joint International Conference on Information Sciences*. Atlantis Press, 2008.

[27] A. Beer and R. Ramler, "The role of experience in software testing practice," in *34th Euromicro Conference Software Engineering and Advanced Applications*. IEEE, 2008, pp. 258–265.

[28] J. Itkonen, M.V. Mäntylä, and C. Lassenius, "The role of the tester's knowledge in exploratory software testing," *IEEE Transactions on Software Engineering*, Vol. 39, No. 5, 2013, pp. 707–724.

[29] O. Taipale, K. Karhu, and K. Smolander, "Observing software testing practice from the viewpoint of organizations and knowledge management," in *First International Symposium on Empirical Software Engineering and Measurement*. IEEE, 2007, pp. 21–30.

[30] L. Xue-Mei, G. Guochang, L. Yong-Po, and W. Ji, "Research and implementation of knowledge management methods in software testing process," in *WRI World Congress on Computer Science and Information Engineering*, Vol. 7. IEEE, 2009, pp. 739–743.

[31] B.A. Kitchenham, T. Dyba, and M. Jorgensen, "Evidence-based software engineering," in *Proceedings of the 26th International Conference on Software Engineering*. IEEE Computer Society, 2004, pp. 273–281.

[32] J. Webster and R.T. Watson, "Analyzing the past to prepare for the future: Writing a literature review," *MIS quarterly*, 2002, pp. xiii–xxiii.

[33] W. Hayes, "Research synthesis in software engineering: a case for meta-analysis," in *Sixth International Software Metrics Symposium*. IEEE, 1999, pp. 143–151.

[34] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE '14. New York, NY, USA: ACM, 2014, pp. 38:1–38:10. [Online]. http://doi.acm.org/10.1145/2601248.2601268

[35] B. Kitchenham, O.P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering – a systematic literature review," *Information and software technology*, Vol. 51, No. 1, 2009, pp. 7–15.

[36] B. Kitchenham, R. Pretorius, D. Budgen, O.P. Brereton, M. Turner, M. Niazi, and S. Linkman, "Systematic literature reviews in software engineering – a tertiary study," *Information and Software Technology*, Vol. 52, No. 8, 2010, pp. 792–805.

[37] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele University & University of Durham, EBSE Technical Report EBSE 2007-01, 2007.

[38] C.W. Knisely and K.I. Knisely, *Engineering communication*. Cengage Learning, 2014.

[39] D.S. Cruzes and T. Dybå, "Research synthesis in software engineering: A tertiary study," *Information and Software Technology*, Vol. 53, No. 5, 2011, pp. 440–455.

[40] C. Goulding, *Grounded theory: A practical guide for management, business and market researchers*. Sage, 2002.

[41] R. Hoda, J. Noble, and S. Marshall, "Using grounded theory to study the human aspects of software engineering," in *Human Aspects of Software Engineering*. ACM, 2010, p. 5.

[42] B.G. Glaser, A.L. Strauss, and E. Strutzel, "The discovery of grounded theory; strategies for qualitative research." *Nursing research*, Vol. 17, No. 4, 1968, p. 364.

[43] M. Dixon-Woods, S. Agarwal, D. Jones, B. Young, and A. Sutton, "Synthesising qualitative and quantitative evidence: a review of possible methods," *Journal of Health Services Research & Policy*, Vol. 10, No. 1, 2005, pp. 45–53.

[44] D. Koznov, V. Malinov, E. Sokhransky, and M. Novikova, "A knowledge management approach for industrial model-based testing," in *Proceedings of the International Conference on Knowledge Management and Information Sharing*, 2009, pp. 200–205.

[45] M. Rodgers, A. Sowden, M. Petticrew, L. Arai, H. Roberts, N. Britten, and J. Popay, "Testing methodological guidance on the conduct of narrative synthesis in systematic reviews: effectiveness of interventions to promote smoke alarm ownership and function," *Evaluation*, Vol. 15, No. 1, 2009, pp. 49–73.

[46] M. Ivarsson and T. Gorschek, "A method for evaluating rigor and industrial relevance of tech-

nology evaluations," *Empirical Software Engineering*, Vol. 16, No. 3, 2011, pp. 365–395.

[47] A. Jonsson and G. Svingby, "The use of scoring rubrics: Reliability, validity and educational consequences," *Educational Research Review*, Vol. 2, No. 2, 2007, pp. 130–144.

[48] B. Moskal, K. Miller, and L. King, "Grading essays in computer ethics: rubrics considered helpful," *ACM SIGCSE Bulletin*, Vol. 34, No. 1, 2002, pp. 101–105.

[49] A. Vickers, "Ensuring scientific rigour in literature review," *Acupuncture in Medicine*, Vol. 13, No. 2, 1995, pp. 93–96.

[50] P. Runeson, M. Host, A. Rainer, and B. Regnell, *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons, 2012.

[51] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, "Requirements engineering paper classification and evaluation criteria: A proposal and a discussion," *Requirements Engineering*, Vol. 11, No. 1, 2006, pp. 102–107.

[52] P. Goodman Jon and C. Huseman Richard, *Leading with Knowledge: The Nature of Competition in the 21st Century*. Sage, London, 1999.

[53] L. Rajiv and M. Sarvary, "KM and competition in the consulting industry," 1999, p. 485.

[54] R. de Almeida Falbo, "Experiences in using a method for building domain ontologies," in *The 16th International Conference on Software Engineering and Knowledge Engineering, SEKE*, 2004, pp. 474–477.

[55] S. Vegas and V. Basili, "A characterisation schema for software testing techniques," *Empirical Software Engineering*, Vol. 10, No. 4, 2005, pp. 437–466.

[56] A. Abran, P. Bourque, R. Dupuis, and J.W. Moore, *Guide to the software engineering body of knowledge – SWEBOK*. IEEE Press, 2001.

[57] M. Cataldo, P.A. Wagstrom, J.D. Herbsleb, and K.M. Carley, "Identification of coordination requirements: implications for the design of collaboration and awareness tools," in *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work*. ACM, 2006, pp. 353–362.

[58] D. Graham, E. Van Veenendaal, and I. Evans, *Foundations of software testing: ISTQB certification*. Cengage Learning EMEA, 2008.

[59] S. Vegas, N. Juristo, and V.R. Basili, "A process for identifying relevant information for a repository: A case study for testing techniques," in *Managing Software Engineering Knowledge*. Springer, 2003, pp. 199–230.

[60] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[61] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering." in *EASE*, Vol. 8, 2008, pp. 68–77.

[62] C. Kaner and W.P. Bond, "Software engineering metrics: What do they measure and how do we know?" in *In METRICS 2004. IEEE CS*. Citeseer, 2004.

[63] E.F. de Souza, R. de Almeida Falbo, and N.L. Vijaykumar, "Knowledge management applied to software testing: A systematic mapping," in *The 25th International Conference on Software Engineering and Knowledge Engineering, SEKE*, Boston, USA, 2013, pp. 562–567.

[64] A. Tinkham and C. Kaner, "Learning styles and exploratory testing," in *Proceedings of the Pacific Northwest Software Quality Conference*, 2003.

[65] S. Vegas, N. Juristo, and V. Basili, "Packaging experiences for improving testing technique selection," *Journal of Systems and Software*, Vol. 79, No. 11, 2006, pp. 1606–1618.

[66] J. Itkonen, M.V. Mantyla, and C. Lassenius, "How do testers do it? An exploratory study on manual testing practices," in *Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement*. IEEE Computer Society, 2009, pp. 494–497.

[67] A. Freitas and R. Vieira, "An ontology for guiding performance testing," in *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Vol. 1. IEEE Computer Society, 2014, pp. 400–407.

[68] T.E. Lee, "Applying knowledge management approach for software testing," in *Advances and Innovations in Systems Testing*, 2007.

[69] E.F. de Souza, R. de Almeida Falbo, and N.L. Vijaykumar, "Ontologies in software testing: A systematic literature review," in *VI Seminar on Ontology Research in Brazil*, 2013, p. 71.

[70] C. Kerkhof, J. van den Ende, and I. Bogenrieder, "Knowledge management in the professional organization: a model with application to CMG software testing," *Knowledge and Process Management*, Vol. 10, No. 2, 2003, pp. 77–84.

[71] S. Vegas, N. Juristo, and V.R. Basili, "Maturing software engineering knowledge through classifications: A case study on unit testing techniques," *IEEE Transactions on Software Engineering*, Vol. 35, No. 4, 2009, pp. 551–565.

[72] R. Gentry and F. Shirazi, "A knowledge management analysis of an in-house manual software testing," *International Journal of Computer Application*, Vol. 1, No. 5, 2015, pp. 13–37.

[73] E.F. de Souza, R. de Almeida Falbo, and N.L. Vijaykumar, "Using ontology patterns for building a reference software testing ontology," in *17th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW)*. IEEE, 2013, pp. 21–30.

[74] V.H. Nasser, W. Du, and D. MacIsaac, "Knowledge-based software test generation." in *The 21st International Conference on Software Engineering and Knowledge Engineering, SEKE*, 2009, pp. 312–317.

[75] H. Li, H. Guo, F. Chen, H. Yang, and Y. Yang, "Using ontology to generate test cases for GUI testing," *International Journal of Computer Applications in Technology*, Vol. 42, No. 2-3, 2011, pp. 213–224.

Appendix A. Publication venue for the selected papers

Table A. Publication venue for the selected papers

Publication	Type	ID
Malaysian Conference in Software Engineering (MySEC)	Conference	P1 [5]
Information and Software Technology	Journal	P2 [3]
International Conference on Information Technology and Computer Science	Conference	P3 [4]
International Conference and Workshop on Emerging Trends in Technology (ICWET)	Conference	P4 [6]
International Conference on Industrial Engineering and Engineering Management	Conference	P5 [14]
Information and Software Technology	Journal	P6 [1]
International Symposium on Empirical Software Engineering and Measurement	Conference	P7 [29]
International Conference on Internet Computing for Science and Engineering (ICICSE)	Conference	P8 [9]
WRI World Congress on Computer Science and Information Engineering	Conference	P9 [30]
Euromicro Conference on Software Engineering and Advanced Applications	Conference	P10 [27]
IEEE Transactions on Software Engineering	Journal	P11 [28]
Journal of Workplace Learning	Journal	P12 [7]
International Conference on Knowledge Management and Information Sharing	Conference	P13 [44]
Empirical Software Engineering	Journal	P14 [55]
International Symposium on Empirical Software Engineering, ISESE	Conference	P15 [25]
International Conference on Software Engineering and Knowledge Engineering	Conference	P16 [63]
Pacific Northwest Software Quality Conference	Conference	P17 [64]
Journal of Systems and Software	Journal	P18 [65]
Joint International Conference on Information Sciences	Conference	P19 [26]
International Symposium on Empirical Software Engineering and Measurement	Conference	P20 [66]
International Performance Computing and Communications Conference (IPCCC)	Conference	P21 [15]
International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)	Conference	P22 [67]
Information Systems Research	Conference	P23 [68]
Seminar on Ontology Research in Brazil	Conference	P24 [69]
Journal of Software Engineering and Applications	Journal	P25 [16]
International Conference on Software Engineering and Knowledge Engineering	Conference	P26 [17]
Knowledge and Process Management	Journal	P27 [70]
Empirical Software Engineering	Journal	P28 [20]
International Conference on Software Engineering and Knowledge Engineering	Conference	P29 [19]
Software Engineering Journal	Journal	P30 [71]
International Journal of Computer Application	Journal	P31 [72]
International Enterprise Distributed Object Computing Conference Workshops (EDOCW)	Conference	P32 [73]
International Conference on Software Engineering and Knowledge Engineering	Conference	P33 [74]
International Journal of Computer Applications in Technology	Journal	P34 [75]
Computer Software and Applications Conference (COMPSAC)	Conference	P35 [18]

Appendix B. Quality assessment based on rigor and relevance

Table B. Quality assessment based on rigor and relevance

Paper	Context	Study design	Validity	Rigor sum	Subjects	Scale	Research method-ology	Context	Relevance sum
P1	1	1	0	2	1	1	1	1	4
P2	1	1	1	3	1	1	1	1	4
P3	1	0.5	0	1.5	1	1	1	1	4
P4	1	0.5	0	1.5	1	1	1	1	4
P5	1	1	0	2	1	1	1	1	4
P7	1	1	1	3	1	1	1	1	4
P8	1	0.5	0	1.5	1	1	1	1	4
P9	1	0.5	0	1.5	1	1	1	1	4
P10	1	1	0	2	1	1	1	1	4
P11	1	1	1	3	1	1	1	1	4
P12	1	0.5	0	1.5	1	1	1	1	4
P13	0.5	0.5	0	1	1	1	1	1	4
P14	1	1	0	2	1	1	1	1	4
P15	1	1	0	2	1	1	1	1	4
P17	0	0.5	0	0.5	0	1	0	0	1
P18	1	0.5	0	1.5	0.5	1	0	1	2.5
P19	0.5	0.5	0	1	1	1	1	1	4
P20	1	1	0	2	1	1	1	1	4
P21	0	0.5	0	0.5	0	1	1	0	2
P22	0	0.5	0	0.5	0	1	1	0	2
P23	0.5	0	0	0.5	0	1	1	0	2
P25	1	1	0	2	1	1	1	1	4
P26	0	0.5	0	0.5	0	1	1	0	2
P27	0.5	0.5	0	1	1	1	1	1	4
P28	0.5	0.5	0	1	1	0.5	0.5	1	3
P29	0	0.5	0	0.5	1	1	1	1	4
P30	1	1	0	2	1	1	1	1	4
P31	1	1	1	3	1	1	1	1	4
P32	0	0.5	0	0.5	0	1	1	0	2
P33	1	0.5	0	1.5	1	1	1	1	4
P34	1	0.5	0	1.5	1	1	1	1	4
P35	1	1	0	2	1	1	1	1	4

e-Informatica Software Engineering Journal, Volume 12, Issue 1, 2018, pages: 79–115, DOI 10.5277/e-Inf180104

Tool Features to Support Systematic Reviews in Software Engineering – A Cross Domain Study

Chris Marshall*, Barbara Kitchenham**, Pearl Brereton**
* *York Health Economics Consortium Ltd., University of York*
** *School of Computing and Mathematics, Keele University*
chris.marshall@york.ac.uk, b.a.kitchenham@keele.ac.uk, o.p.brereton@keele.ac.uk

Abstract

Context: Previously, the authors had developed and evaluated a framework to evaluate systematic review (SR) lifecycle tools.
Goal: The goal of this study was to use the experiences of researchers in other domains to further evaluate and refine the evaluation framework.
Method: The authors investigated the opinions of researchers with experience of systematic reviews in the healthcare and social sciences domains.
They used semi-structured interviews to elicit their experiences of systematic reviews and SR support tools.
Results: Study participants found broadly the same problems as software engineering (SE) researchers with the SR process. They agreed with the tool features included in the evaluation framework. Furthermore, although there were some differences, the majority of the importance assessments were very close.
Conclusions: In the context of SRs, the experiences of researchers in other domains can be useful to software engineering researchers. The evaluation framework for SR lifecycle tools appeared quite robust.

Keywords: software engineering, systematic review tools, cross-domain survey, qualitative analysis

1. Introduction

A systematic review (SR) is a formal, repeatable method for identifying, evaluating and interpreting all available research regarding a particular problem or topic of interest. The rigorous and impartial nature of a systematic review increases the scientific value of its findings in comparison with expert-based literature reviews [1–3], which makes it an important tool for obtaining and appraising evidence in a reliable, transparent and objective way. Systematic reviews were first established in Clinical Medicine [4, 5]. Medical researchers defined the systematic review process to help mitigate the drawbacks of a conventional literature review [1]. A cautionary note needs to be added here that systematic reviews have received some criticism, in particular, that they are sometimes of quite poor quality and can reap high rewards in terms of citation counts despite biases and vested interests [6]. Also, the synthesis of outcomes, particularly in the software engineering field, can be problematic [7].

With a growing emphasis on empirical software engineering research, the popularity and importance of systematic reviews has grown considerably [8, 9]. Despite their potential usefulness and importance to empirical software engineering research, undertaking a systematic review remains a highly manual and labour intensive process resulting in the possibility of process errors (such as misclassifying primary studies or wrongly excluding a primary study). In particular, there are challenges concerning the study selection, data extraction and data synthesis stages, amongst other collaborative activities

[10–14]. Furthermore, systematic reviews have only recently been adopted by software engineering researchers, and, as a result, there have been problems surrounding the provision of appropriate support for novices [11–14]. These drawbacks, along with others, make the systematic review methodology a prime candidate to benefit from an automated tool support [12–16].

In our experience, it is certainly possible to undertake a systematic review without too much automation. Furthermore, Kitchenham and Brereton were involved in the revision of the systematic review guidelines that emphasised human processes and decision making [17]. Thus, the authors believe it is important to have a balanced view of the benefits of automating the systematic review process. In this study, attitudes to automation in domains that have more practical experience of systematic reviews and their automation than software engineering were investigated.

In earlier research, the authors developed and validated a framework for evaluating tools intended to support the full systematic review process [18]. The framework was based on a set of tool features identified as important for systematic reviews in software engineering based on the SR guidelines, the authors experiences, and the experiences of other SE researchers reported in the literature. This paper reports on the results of a cross-domain study of researchers who undertake systematic reviews as part of their normal research practice, which was intended to further validate our framework.

Some of this research has already been reported [19], however, this paper provides a more detailed analysis of our study results relating to the impact of participant's experience level and the identification of trends among their comments (the additional analyses are itemized in Section 4.2.3).

Section 2 describes the evaluation framework and explains particular interest in systematic review lifecycle tools. Section 3 discusses SE research that used results from other disciplines, that investigated benefits and problems with the SR process, and discussed tools to support the SR process. Section 4 discusses the goals of the study and the methodology used to address these goals. Section 5 presents the results of the cross domain study. Section 6 discusses the results and conclusions are presented in Section 7.

2. Framework for evaluating systematic review lifecycle tools

The developed evaluation framework was aimed at evaluating tools that support the full SR process in contrast to tools that assist a specific process or task. The reasons why the authors concentrated on these tools and developed a multi-criteria decision making framework are:

1. Large SRs are complex and hard to manage. In order to support the production and update of large scale (possibly distributed) SRs, standard tools such as reference managers and spread sheets become increasingly cumbersome and error prone. The developers of the SLuRp tool say "Our experience is that in order to produce reliable valid results, more than one reviewer is required. Maintaining large amounts of data in a team with several reviewers is time-consuming and error-prone. These errors are difficult to identify and eliminate without the use of a specific SLR tool like SLuRp." [20].
2. SR lifecycle tools cannot be easily evaluated. Tools that support a specific process or task can be evaluated in isolation using experiments or small case studies, in contrast SR lifecycle tools are more difficult to evaluate because they span the entire lifecycle of a review from initial planning to final reporting and even subsequent updating. This lifecycle process is made up of a series of individual processes that interact with one another and require validation and sometimes reworking. To maintain clarity within this paper we shall refer to these tools as SRLC (Systematic Review LifeCycle) tools.
3. Currently, there is interest among software engineering research groups in building SRLC tools. The initial search found four such tools [21] and later another one was found [22]. This interest suggests it is an appropriate time to consider how to evaluate such tools.

4. Adopting such tools is a major commitment. Research groups need to have some confidence that any tool they adopt will be able to support the sort of systematic reviews they perform and the way in which they manage their systematic review process.

The evaluation framework was based on feature analysis as proposed by the DESMET project [23]. Feature analysis is a type of multi-criteria decision analysis. It is a subjective method of evaluation. It is intended to provide a means of organising a subjective evaluation of a tool and making the components of that evaluation clear to, and auditable by other potential tool users.

In the context of SRLC tools, members of the same software engineering research group were expected to be other potential users. Thus, the authors envisage that our framework would provide a means by which researchers could make an informed, defensible decision together. One particular benefit of the DESMET feature analysis method is that it requires the users of the method to refine the evaluation process depending on their own requirements. Specifically it involves users of the feature analysis defining what they require of an acceptable tool with respect of each feature. So the users of the framework do not just evaluate a tool against a set of features, they also need to define the importance of each feature in terms of its importance to them. This means that although an evaluation exercise could involve a series of different candidate SRLC tools, the tools are not so much compared with each other as with the research group's specific set of requirements. This provides a feature analysis with a built-in element of flexibility, which allows users to tailor an evaluation to their own circumstances. The details of the initial version of the framework and its evaluation can be found in [18].

3. Related work

In 2004, Kitchenham et al. [24] introduced the concept of Evidence-Based Software Engineering (EBSE) as an approach to integrate academic research with industry needs and improve decision making regarding the development and maintenance of software. This initiative was based on the concept of Evidence-Based Medicine. Kitchenham et al. recommended the use of systematic reviews to support EBSE. Subsequently, Kitchenham [25] developed a set of guidelines for undertaking systematic reviews based on health care guidelines, which were updated in 2007 [3]. The 2007 guidelines were influenced both by a study of the use of systematic reviews in other disciplines and by guidelines developed for the social sciences [26], and were adapted to better reflect the use of systematic reviews in software engineering. A further update to the guidelines was released in 2015 (see Section III of [17]). This version of the guidelines was strongly oriented to addressing software engineering issues. In particular, it included more information about managing the collaboration aspects of systematic reviews and methods for synthesizing the results of quantitative and qualitative studies.

Since the release of the original guidelines and the publication of systematic reviews in software engineering journals, there has been substantial literature discussing how the software engineering community performs systematic reviews and how the process could be made more efficient. Kitchenham and Brereton [9] summarized this literature in a systematic review that included 45 papers published between January 2005 and June 2012. This study summarized the perceived benefits of doing SRs, problems SE researchers had found when undertaking SRs and the advice and techniques intended to assist in performing SR tasks. However, most of this work was fairly inward looking with relatively few papers discussing ideas from outside the software engineering community. The main exceptions were: Torres et al. [27] who trialled the methods of sentence classification used in scientific papers on SE data; Felizardo et al. [28] who undertook a cross-discipline mapping study to investigate the use of visual data mining techniques to support SRs; Ramampiaro et al. [16] who discussed the use of techniques from information retrieval and text mining to support the development of meta-searcher capabilities.

Since 2012, there have been two initiatives to investigate tools to support systematic reviews in software engineering undertaken independently by two groups of researchers:

1. Marshall and Brereton [21] performed a mapping study to identify tools available to support SRs in the SE community and identified 13 different tools of which three were intended to support the full lifecycle (i.e. were SRLC tools). They also introduced the systematic review toolbox which is a catalogue of tools to support systematic reviews [29]. All three authors of this paper presented an evaluation framework intended to assess SRLC tools and reported the results of using the evaluation framework to evaluate four different SRLC tools developed in the software engineering community [18]. They also published a preliminary analysis of data from our study of researchers in health care and social science [19].
2. Carver et al. [14] reported barriers to the SR process based on 52 responses to an online survey sent to authors who published SRs in SE venues and qualitative experiences from eight PhD students. Hassler et al. [30] reported the result of a community workshop that identified and ranked 37 barriers to the SR process that could be grouped into themes related to the SR process, primary studies, the practitioner community and tooling. Subsequently, Hassler et al. [31] reported a workshop-based study of SR tool needs based on information provided by 16 software engineering researchers. They compared the result of their study with the published preliminary results of our study of tool features [19].

4. Goals and methodology of the cross-domain study

4.1. Goals

The objective was to see if the experiences of researchers from domains that have more extensive experience in the use of systematic reviews would be valuable to software engineering (SE) researchers and SR tool designers. In particular, the goals of this study were:

1. To assess whether the SR experiences of researchers in other domains are relevant to those of SE researchers.
2. To explore what tools were currently available and used to support systematic reviews in other domains.
3. To compare the features and importance levels identified by the participants with those in this SRLC tool evaluation framework.

These goals could best be addressed by a qualitative study aimed at eliciting the experiences of systematic reviewers on other domains. For this reason, Marshall undertook a series of cross-domain, semi-structured interviews, which were designed to explore the experiences and opinions of systematic reviewers in other domains (outside of software engineering) about support tools.

It should be noted that, as is common with qualitative studies, the goals are fairly general and do not map to detailed research questions and hypotheses. They exist to scope the qualitative study not to define questions and metrics.

4.2. Methodology of the cross-domain study

This section reports on the research strategy and research process.

4.2.1. Research strategy

Semi-structured interviews were used to elicit the opinions of researchers about systematic review support tools. This means that a number of questions were identified to ask the participants and also to encourage a discussion about the issues to follow the directions that the participants wanted. Semi-structured interviews were selected instead of a self-administered questionnaire for two main reasons:

1. The awareness that terminology differs between different domains and that face-to-face interviews would allow potential misunderstandings to be identified and resolved.
2. The need for certainty that the identified participants had appropriate experience.

Since the study was qualitative, no detailed research questions or research hypotheses were

derived, data collection and analysis procedures arose from the research goals and resulted from the expectations that:

- Viewpoints of researchers working in domains where systematic reviews are well-understood and considered a standard research practice would be valuable to software engineering researchers.
- Viewpoints of novices and experts would differ.
- Tool feature preferences of participants would be influenced by the type of systematic review they undertook.

Thus, the selected study participants covered various domains, different levels of researcher experience and different systematic review types. The aim was to interview both senior and junior researchers from several different domains. Originally, six topic areas were considered: Clinical Medicine, Criminology, Education, Empirical Psychology, Nursing & Midwifery, and Primary Care, however, in practice two high level domains became the focus: social sciences and health care. No restriction was placed on whether the researchers had performed quantitative or qualitative reviews. The goal was to interview researchers with experience of both types of review because issues related to data extraction and aggregation are very different for qualitative and quantitative reviews.

The inclusion criteria for participants were as follows:

- Researchers used systematic reviews as part of their standard research process.
- Researchers had a wide range of roles and responsibilities.

Initially it was planned to provide a *theoretical sample* covering the six topic areas. The theoretical sample is a type of purposeful sampling where researchers are seeking incidents/reports of the phenomenon they are studying which will supply useful data [32]. However, after the data was collected and tabulated, it was found out that the coverage of three dimensions had been achieved:

- The two domains (health care and social sciences).
- Three experience levels corresponding to 1–5 SRs (i.e. Low), 6–15 SRs (i.e. Medium), and > 15 SRs (i.e. High)[1].
- Types of SRs performed: Quantitative and Qualitative.

This coverage of three important dimensions allowed to extend the analysis of the study results.

4.2.2. Research process

Marshall developed the semi-structured interview plan after discussions with Kitchenham and Brereton. He, then, piloted the semi-structured interview procedure with a PhD student who had undertaken two SRs. This led to some changes to the delivery and sequencing of questions and also confirmed the expectation that interviews would take approximately 45 minutes. The interview plan included questions related to four concerns:

- Group 1: questions relating to the participant's background and domain.
- Group 2: questions about the participant's experience of undertaking systematic reviews.
- Group 3: questions about the participant's use of systematic review tools.
- Group 4: questions about SRLC tool features and their importance levels.

The detailed interview questions are reported in Appendix A.

In the research a combination of convenience and snowballing sampling techniques was used to identify 49 potential participants. Finally, 13 researchers from six institutions agreed to take part. Marshall carried out the interviews between June 2014 and September 2014. Prior to the interview, each participant was sent an Interview Preparation Form (see Appendix B). This document outlined the main themes to be covered during the interview, the expected duration, and measures which would be taken to ensure privacy and confidentiality. All interviews were carried out face-to-face and recorded using a digital audio recorder. Marshall took notes throughout each interview. The shortest interview took 32 minutes and the longest interview lasted 68 minutes, with an average of 45 minutes.

[1]For some analyses, only two experience levels were used: low corresponding to 1–5 SRs and high corresponding to 6+ SRs, giving us six relative novices and 7 relatively highly experienced participants.

Marshall processed the raw data (i.e. recordings, field notes) prior to analysis. The field notes were reviewed and full transcriptions of each interview were produced. For this study, transcripts aimed to reflect a straightforward summary of the main ideas, which were presented by a fluently spoken participant. The transcripts did not include any mispronunciations, pauses or word emphases which might have occurred during the interview. In total, the interviews generated approximately 10 hours of audio recordings, each taking between five and six hours to fully transcribe.

4.2.3. Data analysis

Marshall conducted the initial analysis concurrently with data collection, as recommended by Miles et al. [33]. The initial analysis was based on tabulating responses in order to identify:

- Challenges participants faced when doing systematic reviews.
- Tools used by participants.
- Positive and negative experiences of tools.
- Participant opinions of the importance of the features included in the evaluation framework compared with the importance assigned to them.

Kitchenham and Brereton reviewed all the tables for consistency. Initially, comments were tabulated verbatim (as reported in [19]). Subsequently, all three authors reviewed the initial analyses and realized from the biographical data that the actual sample included participants with a range of experiences that would enable additional analyses of the data. This resulted in Kitchenham and Brereton undertaking additional analyses (beyond those reported in [19]) that are reported in this paper and which are described below:

1. A summary of the general problems/issues reported by participants and cross-referenced to the SE literature in order to identify similarities and differences between the SE domain and health care and social services domains.
2. An analysis of the comments by individual participants concerning general systematic review tools and systematic review lifecycle tools. This was intended to give a balanced view of the advantages and disadvantages of automating the SR process.
3. A thematic analysis of the comments related to systematic review lifecycle tool features to provide some quantification of trends. Details of the coding process and an example of how the codes were established is provided in Appendix C.
4. An investigation of whether participants' responses were influenced by their experience of undertaking systematic reviews.
5. An investigation of whether participants' responses were influenced by the type of systematic review they performed.
6. An investigation of the importance of factors related to the usability and ease of installation. This was intended to clarify the features required to represent tool usability.
7. A comparison of our results with other related SE studies. This was intended to highlight similarities and differences between the SE domain and health care and social services domains, particularly in the context of participant experience.

5. Results of the cross-domain study

The details of the participants' roles, research domains and SR experience are given in Table 1. The participants covered a range of disciplines, including nursing, psychology and education in the domains of health care and social sciences, and a variety of roles, including research associate[2], lecturer, senior lecturer[3], information officer/specialist and professor. The term information officer/specialist is used to identify someone whose main role is to provide support for the search process of systematic reviews. This job title confirms the importance of systematic reviews in the health care and social sciences domains.

The group of 13 participants in this study had experience of different types of a system-

[2]Usually a post-doctoral researcher working on a funded project and employed on a fixed-term contract.
[3]An academic position in the UK corresponding to an Associate or Assistant Professor in the USA.

Table 1. Cross domain study participant information

ID	Role	Domain	No. of SRs	Type of SR
P01	Research Associate	Health care (Primary Care)	6–10 (Medium)	Both
P02	Research Associate	Health care	1–5 (Low)	Quantitative
P03	PhD Student	Health care (Physiotherapy)	1–5 (Low)	Qualitative
P04	Senior Lecturer	Health care (Health Psychology)	1–5 (Low)	Qualitative
P05	Information Officer	Health care	11–15 (Medium)	Quantitative
P06	Lecturer	Health care (Nursing)	1–5 (Low)	Quantitative
P07	Lecturer	Social Science (Educational Psychology)	1–5 (Low)	Quantitative
P08	Information Officer	Social Science	> 15 (High)	Both
P09	Professor	Social Science	> 15 (High)	Both
P10	Systematic Reviewer	Social Science (Public Health)	6–10 (Medium)	Both
P11	Research Associate	Social Science (Education Technology)	1–5 (Low)	Both
P12	Professor	Social Science (Education & Child Psychology)	> 15 (High)	Qualitative
P13	Information Specialist	Health care	> 15 (High)	Both

atic review, different levels of experience, and different domains of interest. Specifically:

- In the health care domain, there were seven participants; two concentrated on qualitative reviews, three on quantitative reviews, and two conducted both types of review. Four of the participants were relative novices who had conducted 1–5 reviews, but of the remaining three, one had performed 6–10 reviews, one 11–15 reviews and one > 15 reviews.
- In the social science domain, there were six participants; one concentrated on qualitative reviews, one on quantitative reviews and four conducted both types of reviews. Two of the participants were relative novices (1–5 reviews), one had conducted 6–10 reviews and three had conducted > 15 reviews.

Thus, there was a good coverage of the factors expected to influence the participants' responses in these semi-structured interviews: domain, experience and type of review.

5.1. Issues faced by researchers in other domains

An important issue when evaluating the participants' answers was to determine whether their experiences were relevant to software engineering researchers. In order to investigate this issue the participants were asked about the main challenges and specific problems they had faced when conducting systematic reviews.

Table 2 summarizes the challenges and issues mentioned by the participants. In columns three and four, it was identifies whether these issues had been raised in the SE literature. Column 3 refers to issues that are general problems and identifies whether they are raised in [9] or in [14]. Column 4 refers to process factors discussed in the recent SE related text book which [17] includes an update of guidelines for systematic reviews in software engineering. Column 5 identifies the participants who made a comment and Column 6 specifies their experience.

Table 2 identifies three high level concerns (i.e. those unrelated to specific SR activities) that were mentioned 11 times by six different participants. It is interesting that none of those participants had the highest level of experience. Possibly after doing many SRs, researchers overcome their initial perception of the difficulty of SRs, or, in the case of perceiving SRs to be *Time Consuming*, become inured to the issue.

In the case of the challenges related to specific SR processes, Management issues produced the most comments, both in terms of unique issues raised (of which there were seven), and in terms of the total number of comments (of which there were 13) which were made by eight different participants. It is interesting that the SE literature on SR challenges summarized by

Table 2. Challenges and specific issues reported in interviews

Main Challenges	Interview Specific Issues	Discussed in [9] or [14]	Discussed in [17]	Id	Experience
Search Process	Search String translation	Yes	No	P01	M
	Inconsistency with terminology	Yes	No	P01, P06, P09, P10	M, L, H, M
	Time consuming	Yes	No	P03	L
	Developing the search strategy	No	Yes	P04, P08, P10, P13	L, H, M, H
Time consuming	General	Yes	No	P02, P03, P04, P05, P07, P11	L, L, L, M, L, L
No Standardization	General	Yes	No	P02	L
High Difficulty	General	Yes	No	P02, P03, P07, P11	L, L, L, L
Management	Managing large-scale SRs	No	Yes	P04, P05, P09	L, M, H
	Transparency	No	Yes (reporting)	P05	M
	Handling duplicates	Yes	Yes	P06, P07	L, L
	Collaboration	Yes	Yes	P06, P07, P12, P13	L, L, H, H
	Negotiating with policy makers	No	No	P10	L
	Relationships between studies & papers	No	Yes	P12	H
	Version control	No	No	P12	H
Analysis	Qualitative Analysis	Yes	Yes	P05	H
	Meta-analysis	No	Yes	P06, P10	L, M
Study selection & screening	Resolving disagreements	Yes	Yes	P06	L
	Managing the criteria	Yes	Yes	P12	H
	Criteria consistency across multiple coders	Yes	Yes	P12, P13,	H, H
	General	Yes	Yes	P05, P08	M, H
Quality assessment & critical appraisal	Resolving disagreements	No	Yes	P06	L
	Managing the criteria	Yes	Yes	P12	H
	Criteria consistency over multiple coders	Yes	Yes	P12, P13	H, H
	Assessing quality of study not the paper	Yes	Yes	P12	H
	General	Yes	Yes	P11	L
Protocol Development	Developing research questions	Yes	Yes	P08, P10	H, M
	General	Yes	Yes	P10	M
Producing Report	Formatting references	No	No	P13	H
	General	No	Yes	P10	M
Validation	Knowing when to check for consistency	No	Yes	P12	H

Kitchenham and Brereton [9] did not concentrate on these issues, although they feature more extensively in Hassler et al. [31] and in the latest SR guidelines [17]. This might reflect the greater maturity in the health care and social sciences domains and allows to identify an area which will become more important for SE researchers in the future. Other activities that attracted numerous comments are:

- The search process, with a total of 10 comments about four different issues which were made by eight different participants of all experience levels.
- The study selection and screening process, with a total of six comments consisting of four different issues made by five different participants but including only one comment from a participant with low experience levels.
- The quality assessment and critical appraisal process, with a total of six comments about five different issues made by four participants including two low experience and two high experience participants.

These issues were discussed in the SE literature and the number of high experience participants that mentioned these issues suggests that they remain a challenge irrespective of experience levels.

Three challenges that had no overlap with SE challenges or guidelines are:

1. **Negotiating with policy makers.** Researchers in other domains are often commissioned to do systematic reviews and may, therefore, need to negotiate with the policy makers who commissioned the study. In SE, there are no policy makers who commission systematic reviews, so currently this is not an issue.
2. **Version control.** Systematic reviews in SE are usually considered one-off pieces of research, so are not generally concerned about version control. Researchers in other domains produce reports for policy makers and may need to update those reports periodically, so version control is more important.
3. **Formatting references in the final report.** Although not mentioned as a specific issue in SE papers, it is certainly the case that outputs from different digital libraries are not usually equivalent and can be difficult to integrate, unless converted into an intermediate format compatible with reference manager systems such as EndNote or BibTeX.

These challenges were each mentioned only once.

Overall the results in Table 2 suggest that researchers in other domains face many of the same issues as software engineering researchers. It can be concluded, therefore, that their experiences of tool support for SRs are relevant to those of researchers in software engineering. Furthermore, these results suggest that challenges remain even for highly experienced researchers and, in particular, management issues should be expected to become more important as SE researchers become more experienced. This is likely to happen because as researchers become more experienced with the SR methodology, they will be tempted to take part in more complex and larger scale SRs.

5.2. Tools used in other domains

Table 3 shows the tools that participants reported using to assist their SRs. All but three of the participants (i.e. P10, P11 and P12) reported using reference managers, with RefWorks and EndNote being the most frequently used ones. Six participants used tools that assist analysis including Microsoft Excel, statistical software, meta-analysis tools, and textual analysis tools. Seven participants used SR lifecycle tools: four used RevMan and three used EPPI-reviewer.

Table 4 reports the positive comments participants made about the tools, other than SRLC tools, they used. Both RefWorks and EndNote attracted a large number of positive comments, seven and nine, respectively. However, the comments were generated by three of the four RefWorks users but only two of the five EndNote users.

On the negative side, as shown in Table 5, RefWorks was criticised for its lack of a bulk export feature ("you cannot export all your searches in one go.") and poor usability ("I don't think it's easy to use at all. There are a lot of things compacted onto one screen."). The criticism of EndNote was about whether it could effectively handle large numbers of papers/studies ("people

Table 3. Use of SR lifecycle tools and other tools

ID	SR lifecyle tool	Other tools
P01	RevMan	RefWorks
P02	RevMan	RefMan, STATA, Microsoft Word
P03	None	RefWorks
P04	None	EndNote, NVivo, Microsoft Word
P05	RevMan	RefWorks, Endnote
P06	None	RefWorks, Federated Search Tool
P07	RevMan	Mendeley, Microsoft Excel, Mplus, NVivo, Custom Web-based coding tool, MetaEasy, MetaLight, SPSS
P08	EPPI-reviewer	EndNote, RIS conversion tool
P09	EPPI-reviewer	EndNote, ProCite, Microsoft Word
P10	EPPI-reviewer	None
P11	None	None
P12	None	Microsoft Excel, NVivo, Altal.ti, Mendeley
P13	None	EndNote, Mendeley, PubReMiner, RefMan

are concerned that it doesn't have the capacity to deal with the huge numbers of references.").

Table 6 reports the positive comments about the SRLC tools. The version of EPPI-reviewer current when the interviews took place was EPPI-reviewer 4. It was a comprehensive single or multi-user web-based system for managing systematic reviews across health care and social science domains. During the interviews, the participants were very positive about the variety of ways in which the tool can support the systematic review process (see Table 6). For example, EPPI-reviewer's support for study selection uses text mining to prioritise the most relevant studies, so those are viewed first. It allows the review team to start the full data extraction of the studies before finishing the screening. Its support for thematic analysis uses visualisation techniques to depict the relationships between concepts.

On the negative side, as shown in Table 7, the participants felt EPPI-Reviewer had a steep learning curve ("It's not something you can just pick up and use instantly.") and that it "takes a while to learn all of the different things." In addition, two participants felt that training could be improved.

RevMan primarily supports the preparation and maintenance of Cochrane Reviews, although, it can be used to support other reviews. As can be seen in Table 6, the participants appreciated its good support for statistical analysis techniques, in particular meta-analysis and its support for protocol development.

However, on the negative side some users felt restricted by the tool at times, since some of its features were not accessible unless it was a Cochrane Review ("if your review is not Cochrane commissioned then you can't use that feature of RevMan.") (see Table 7). Other users also felt confused by the tool and felt it was all a bit too complicated.

Both tools exhibit features of particular relevance to the domain they were developed for, i.e. EPPI-reviewer was developed by social scientists and, therefore, provides good support for qualitative analysis. In contrast, RevMan was developed by the Cochrane group primarily to support reviews of randomised controlled trials (RCTs), which are formal medical experiments where experimental subjects are real patients suffering from a specific illness. The reason why RevMan is able to provide support for protocol development is that primary studies should all follow a similar RCT process. Similarly, most RCTs are capable of being synthesized quantitatively, which explains the support for formal meta-analysis.

These results, together with those reported in Table 3, suggest that the users of RevMan may also need to use Reference Manager tools and advanced analysis tools. Although two of the users of EPPI-reviewer reported using other tools, neither reported to need other advanced analysis

Table 4. Participants comments on tools – positives

Tool	Comment	Participant
RefWorks	Okay (Better than doing them by hand)	P01
	Helped manage the search process	P03
	Removes duplicates	P03, P06
	Useful for managing study selection	P03, P06
	Useful for traceability	P03
	Helped share the work load between multiple reviewers	P03
	Useful for handling large numbers of studies	P03
	Able to classify studies using folder	P06
EndNote and EndNote Web	Helps manage the search process	P04, P05
	Links with several databases	P04
	Web-based allowing remote access	P04
	No financial payment required (for EndNote Web)	P04
	Can be used, unconventionally, to support study selection	P04
	Easier to use than RefWorks	P05
	Handles duplicates effectively	P05
	Creates individual databases for each SR project	P05
	Help with search strategy	P05
RefMan	It was OK	P02
Mendeley	Supports collaboration	P07
	Good support for version control	P12
	No financial payment required	P13
Federated search tool	Searches multiple sources	P06
	Useful for piloting search	P06
PubReMiner	Useful for developing protocol	P13
	Helps identify key journals	P13
Custom web-based tool	Supports multiple users (collaboration)	P07
	Exports data into other formats	P07
	Supports role management	P07
STATA	Good usability	P02
	Easier to use than RevMan	P02
NVivo	Helps find themes & trends across papers	P04
MetaEasy	Calculates effect sizes for individual studies	P07
Microsoft Excel	Clear presentation of data	P07
Microsoft Word	Supports protocol development	P02, P04, P09

tools. Furthermore, one user of EPPI-reviewer did not report using any other tool. Thus, it seems that EPPI-reviewer offers more complete support for the systematic review lifecycle than RevMan.

Of the two SRLC tools, EPPI-reviewer is likely to be the most promising one for adoption by software engineers. However, it is possible that it is too much oriented to the requirements of the social sciences domain to be readily usable by software engineering researchers.

5.3. Importance of different features for SRLC tools

Finally, the participants were presented with a list of the features which had included in the evaluation framework for SRLC tools. The par-

Table 5. Participants comments on tools – negatives

Tool	Comment	Participant
RefWorks	Problems with importing search results	P01
	Managing paper-study relationships is confusing	P01
	Not an ideal tool	P03
	Difficult for new users	P03, P06
	Poor usability, user interface	P03, P06
	Lost work	P03, P06
	Difficult to set up	P03
	One database for all reviews – so messy	P05
	Handles duplicates poorly	P05
	Less useful as number of papers increases	P05
	Poor export facility	P05
	Problems formatting references	P06
	Frequent major updates to user interface	P06
	Problems with search engine and database compatibility	P06
EndNote and EndNote Web	Not compatible with all databases	P04
	Extraction can be a bit clunky	P04
	Less useful as number of references increases	P05, P13
	Poor export facility	P05
	Trust issues (Web version is online and free)	P13
RefMan	Unnecessary for small numbers of papers	P02
	Problems formatting references	P13
	Problems with maintenance and support	P13
	Not very effective	P13
	Poor support for collaboration	P13
Mendeley	No version control	P07
	Copyright concerns	P13
Federated search tool	Searches multiple sources	P06
MetaEasy	Poor tool integration	P07
MetaLight	Difficult to use	P07
Microsoft Excel	Not that useful	P07
	No support for version control	P12
	Problems with interface	P12
	Doesn't support complex SR tasks	P12
	Too generic	P12

ticipants were asked to rate the features on a five point ordinal scale:

1. Mandatory – meaning that the feature was essential in any tool aiming to support the SR lifecycle.
2. Highly desirable – meaning that although not mandatory, such a feature is extremely important in a SRLC tool.
3. Desirable – meaning that the feature would be useful for most researchers.
4. Nice-to-have – meaning the feature might be useful, but its omission would not seriously affect the tool's value to its users.
5. Not needed – meaning the feature is unnecessary and there is a danger that the feature would increase the complexity of the tool without adding any useful facilities.

The participants were also asked to identify any important features which had been overlooked.

The counts of the importance ratings of the features given by the 13 participants are presented in Table 8, where the bold number is the modal response rating for the feature.

The points raised by the participants during the discussion of the features are summarized

Table 6. Participants comments on SR lifecycle tools – positives

Tool	Comment	Participant
RevMan	Good support for statistics & meta-analysis	P01, P05
	Support for protocol development	P01
	Nice chart generation	P05
EPPI-reviewer	Supports the whole process	P08
	Good support for study selection	P08, P09
	Supports qualitative analysis (thematic analysis)	P09
	Helps manage the search process	P08, P09
	Generates tables and charts to be used in the report	P08
	Flexible coding system	P09
	Allows data extraction in tandem with study selection	P09
	Exports data into other formats	P09
	Supports basic meta-analysis	P09
	Supports role management	P09
	Customisable interfaces	P09
	Supports re-use of data from past SRs	P09
	Good support for "tedious" bits of SR process	P10
	Good support for document management	P10
	Supports inter-rater reliability	P10
	Easy to use	P10

below. The features relating to the same overall concern are grouped together.

5.3.1. Support for SR tasks

SRLC tool features related to the tasks needed to be performed in a systematic review are labelled SRT1 to SRT11 in Table 8.

Protocol management

Table 9 identifies the main issues participants raised when discussing protocol development and validation. The column labelled Participants identifies the number of participants who made comments related to each of them and the column labelled Experience identifies the experience level of the participants. This table includes the issue referred to a *Viability* which was only mentioned by one person in the context of protocol development and validation. It was included here because it referred to the concern that the feature might not be capable of implementation, which was mentioned by many other participants during discussions of other SR support tools.

With respect to support for developing the review protocol, participants' views differed (see Table 8 row SRT1). Four participants thought it would be used particularly for version control, while two felt it would be useful for complex projects (i.e. large teams). Three participants, however, were unsure of its usefulness since they simply used Microsoft Word to track changes. Another participant pointed out that the Cochrane Handbook assisted with protocol development.

Participants' views also differed with respect to the value of tool support for protocol validation (see Table 8 row SRT2). The two modal responses were Desirable (five participants) and Not needed (five participants). Two participants thought it would help avoid missing anything. However, two other participants felt that introducing automation might be over-complicating the process. In addition, two participants mentioned problems with existing approaches to protocol validation that enforced protocol standards in the context of registering Cochrane reviews and submitting proposals to professional bodies.

Search and study selection

Table 10 displays the main themes related to Search and Study selection. Although none of the participants felt that automated support for the search process was Not needed (see Table 8 row SRT3), the opinions about its importance were

Table 7. Participants comments on SR lifecycle tools – negatives

Tool	Comment	Participant
RevMan	Most features locked out if not doing a Cochrane review	P01,
	Not flexible enough	P02, P07
	Doesn't support many important aspects of SRs	P05
	Limited support for reporting phase	P05
	Confusing	P07
	Over restrictive conceptual model	P07
	Expensive	P07
	Limited support for developing the protocol	P07
	Not nicely integrated	P07
EPPI-reviewer	Problems importing search results	P08
	No support for searching	P08
	Difficult to learn	P09
	Limited training support for novices	P09, P10
	No support for protocol development	P09
	No support for network meta-analysis	P09
	Limited information about updates	P10

Table 8. Importance of features

ID	Feature	Mandatory	Highly desirable	Desirable	Nice	Not needed	Our assessment
SRT1	Protocol development	2	**4**	2	3	2	Desirable
SRT2	Protocol validation	1	1	**5**	1	**5**	Desirable
SRT3	Search process	3	**4**	3	3	0	Highly des.
SRT4	Study selection	5	**6**	2	0	0	Highly des.
SRT5	Quality assessment	5	**7**	1	0	0	Highly des.
SRT6	Data extraction	**7**	5	1	0	0	Highly des.
SRT7	Data synthesis	5	**7**	1	0	0	Highly des.
SRT8	Text analysis	0	3	2	**5**	3	Nice
SRT9	Meta-analysis	4	**5**	2	2	0	Nice
SRT10	Reporting	0	2	**7**	4	0	Nice
SRT11	Report validation	0	3	3	3	**4**	Nice
SRM1	Multiple users	**9**	2	2	0	0	Mandatory
SRM2	Document management	**6**	4	2	1	0	Mandatory
SRM3	Security	**6**	2	1	3	1	Desirable
SRM4	Role management	3	3	2	**4**	1	Highly des.
SRM5	Reuse of past data	3	**7**	3	0	0	N/A
IS1	Ease of setup	**6**	5	1	1	0	Highly des.
IS2	Installation guide	4	**5**	1	3	0	Highly des.
IS3	Tutorial	**4**	**4**	3	2	0	Highly des.
IS4	Self-contained	0	**6**	**6**	0	1	Highly des.
E1	Free	0	**5**	3	1	4	Highly des.
E2	Maintained	6	**7**	0	0	0	Highly des.

Table 9. Comments about protocol development & validation

ID	Feature	Theme	Participants	Experience
SRT1	Protocol development	Helps track changes	2	L(1), H(1)
		Helps version control	4	L(1), M(1), H(2)
		Existing tools	4	L(3), H(1)
		Viability	1	L(1)
		For complex projects	2	H(2)
SRT2	Protocol validation	Bad experiences	2	L(1), H(1)
		Over-complicating things	2	L(1), H(1)
		Useful checklist	2	L(1), H(1)

Table 10. Comments about search & selection

ID	Feature	Theme	Participants	Experience
SRT3	Search process	Time saving	3	L(2), H(1)
		Viability	5	L(2), M(1), H(2)
		Help search strategy	2	M(1), H(1)
SRT4	Study selection	Time saving	3	L(1), M(1) L(1)
		Managing disagreements	3	L(2), H(1)
		Additional checking	2	H(2)

divided among all the other importance levels. Three participants commented that such support would save them a lot of time. However, five participants were concerned that it would be difficult to develop trustworthy automated support (e.g. "It would be highly difficult to automate all that."). Two also mentioned the need for support to help develop the search strategy (e.g. "The bit where our time is most valuable is developing the search strategy in the first place.").

All participants felt that tool support for study selection was useful (see Table 8 row SRT4), with five participants regarding it as Mandatory and six as Highly desirable. Three participants mentioned the potential for saving time. Three thought the facility would be useful for resolving disagreements and two mentioned the opportunity to check that things had not been missed. However, one participant felt that a lot of what the feature was targeting could be solved with a "quick conversation" between the members of the review team.

Quality assessment
and data extraction

Table 11 shows the main themes related to Quality Assessment and Data Extraction. Concerning tool support for quality assessment (see Table 8 row SRT5), the majority of participants felt this would be another useful feature since "all these things otherwise require meetings and organisation". Participants also suggested specific features they would like to see:

- The ability to tailor quality criteria.
- The ability to link the quality assessment to data analysis.
- The ability to compare independent assessments and look for disagreements.

With regards to tool support for data extraction (see Table 8 row SRT6), all participants felt that tool support would be useful, with seven participants regarding it as Mandatory and five as Highly desirable. In the context of an end-to-end tool, one participant said it would make extracted data ready to go "straight into the analysis". Four participants, however, were not sure how such a tool could work particularly when handling qualitative data.

Data analysis and synthesis

Table 12 shows the main themes related to Data Analysis and Synthesis. Concerning automated support for data synthesis (see Ta-

Table 11. Comments about quality assessment & data extraction

ID	Feature	Theme	Participants	Experience
SRT5	Quality assessment	Viability	2	L(1), H(1)
		Managing disagreements	1	H(1)
SRT6	Data extraction	Viability	4	L(3), M(1)

Table 12. Comments about data analysis & synthesis

ID	Feature	Theme	Participants	Experience
SRT7	Data synthesis	Viability	2	L(1), H(1)
		Time saving	3	L(2), H(1)
SRT8	Text analysis	Viability	2	L(2)
		Time saving	1	M(1)
		Managing consistency	1	H(1)
SRT9	Meta-analysis	Not always necessary	4	L(2), M(1), H(1)

ble 8 row SRT7), all participants felt this would be useful, with five suggesting such a feature should be Mandatory and seven suggesting it was Highly desirable. Three participants mentioned potential time saving. One participant felt that "less experienced reviewers would find [this feature] particularly useful". However, two participants mentioned factors that might make such a feature difficult to implement (i.e. many different types of analysis and new analysis methods being ahead of tool support).

Overall support for a text analysis feature was muted (see Table 8 row SRT8); the modal value was Nice-to-have (five participants). Two participants mentioned difficulties implementing such a tool (i.e. missing things and false positives). However, one participant felt that text analysis would become "increasingly more important as the complexity of the literature increases", while another mentioned that the technology was now getting to the stage where such a feature was viable. In terms of possible benefits, one participant thought that it would save time, another that it could be used to check the consistency of reviewers extractions.

The participants felt that tool support for meta-analysis (see Table 8 row SRT9) was either Mandatory (four participants) or Highly desirable (five), although four participants noted that not all SRs require meta-analysis. One participant thought it would be useful for novices as, "for a lot of people undertaking a SR for the first time, meta-analysis is their biggest fear".

Report writing and validation

Table 13 shows the main themes related to report writing and report validation. With a modal value of Desirable, most participants felt that tool support for writing the report was not very important (see Table 8 row SRT10). Three positive comments were that it would give reviewers a starting point. In contrast to this, four participants noted that there are many different formats required by journals, meaning that full support might be unrealistic. Two participants also mention other existing tools (i.e. RevMan for Cochrane reviews and Google Documents).

With regards to tool support for report validation (see Table 8 row SRT11), the modal value was Not needed and the other responses were spread across all the other levels excluding the Mandatory level. Two participants mentioned that there were other existing tools (i.e. Word with track changes and PRISMA).

5.3.2. SR process management

SRLC tool features related to the management of the SR process are labelled SRM1 to SRM5 in Table 8.

Table 13. Comments about report writing & validation

ID	Feature	Theme	Participants	Experience
SRT10	Report writing	Time saving	1	H(1)
		Viability	4	L(3), H(1)
		Starting point	3	L(2), H(1)
		Existing tools	2	L(1), H(1)
SRT11	Report validation	Existing tools	2	L(1), M(1)

Table 14. Comments about SR process management

ID	Feature	Theme	Participants	Experience
SRM1	Multiple users	Multiple-user process	5	L(1), M(2), H(3)
		For complex projects	3	L(2), H1
SRM2	Document management	Document integration	3	M(2), H(1)
SRM3	Security	Already done	2	L(2)
		Proprietary data	5	L(1), M(1), H(3)
SRM4	Role management	Over-complicating things	1	L(1)
		For complex projects	3	L(2), H(1)
		For overseeing	2	M(1), L(1)
SRM5	Re-use	For updates	2	L(1), H(1)
		Use previous work	3	L(1), M(1), H(1)

Table 14 shows the major themes concerning SR process management. The majority of participants felt support for multiple users within a tool was really important with nine participants considering it Mandatory (see Table 8 row SRM1). Five participants noted that people do not write systematic reviews on their own, so such a facility is mandatory. Three participants mentioned it was appropriate for complex projects: one participant thought "It should do for large projects", another "If I was working with people internationally", and another mentioned the SRs are generally "team collaboration type projects".

Most participants felt that tool support for document management would be a useful feature (see Table 8row SRM2), with six participants regarding it as Mandatory and four as Highly desirable. In particular, three participants mentioned the importance of being able to manage links between primary studies and one mentioned "Going from a reference manager to a study-based system".

Most participants felt the feature which supports security, should be included in a tool (see Table 8 row SRM3). Six participants regarded it as Mandatory and two as Highly desirable. Five participants (including one novice) mentioned security was needed to address problems associated with confidential information and intellectual property rights. Two novice participants argued, however, that since SRs deal with published studies, security wouldn't be necessary. It is possible that systematic reviewers with more experience are more likely to have come across reviews where confidentiality was important.

The participants were divided as to the importance of tool support for role management (see Table 8 row SRM4). Although three participants regarded role management as Mandatory, the modal value for this feature was Nice-to-have which was the assessment made by four participants. Three participants felt it was important for complex projects (large teams). Two other participants thought that it would help to get an overview of the whole team, one of them pointing out that it was particularly important for the first author. Another participant, pointed out that "it does not necessarily mean that you don't trust people to do a good job, it would just cut down the chances of a mistake". One novice researcher

Table 15. Comments about ease of use

ID	Feature	Theme	Participants	Experience
IS1	Ease of setup	Depends on tool	2	H(1), H(1)
		Poor installation frustrates	2	M(1), H(1)
		Job for IT staff	2	M(1), H(1)
IS2	Installation guide	Job for IT staff	1	L(1)
IS3	Tutorial	None	n/a	n/a
IS4	Self-contained	Depends on tool	3	L(1), H(2)

mentioned that it might over-complicate the process.

It is possible that systematic reviewers without software engineering experience would not appreciate it that in order to produce a software tool that supports independent quality assessment and data extraction of documents by two or more researchers, it identifies disagreements among their extractions and facilitates the production of a final mediated extraction, a certain kind of role management is essential.

All participants felt that tool support for re-using data from past SRs would be useful (see Table 8 row SRM5). Two participants mentioned it was important for updating existing reviews. Other participants mentioned possible uses of such a feature:

- When using primary studies that were used in a previous SR, the quality assessment could be reused.
- The references for primary studies used in previous SRs would be available.
- Using the search terms, you could automatically identify papers that were used in previous SRs.

5.3.3. Ease of use

Features related to the setup of a SRLC tool are labelled IS1 to IS4 in Table 8.

Most participants were in favour of tools that were easy to setup (see Table 8 row IS1), and included an installation guide (see Table 8 row IS2) and a tutorial (see Table 8 row IS3). They also felt having a self-contained tool[4] was either Highly desirable (six participants) or Desirable (six participants) (see Table 8 row IS4).

Table 15 identifies the main discussion themes for ease of use features, identifying issues that were mentioned more than once. With respect to a simple setup accompanied by an installation guide, three participants mention IT staff were available to handle installation issues. Two participants felt that without a simple installation process, users would become frustrated with a tool. Two participants, however, felt that "if the tool is good enough", then, "some people are prepared to give [the difficult setup] a go". These features are discussed further in Section 5.6.

With respect to whether SR lifecycle tool should be self-contained, three of the participants, felt it was not a really important issue, since they would be quite satisfied to install other packages if the tool "does stuff that nothing else can do".

5.3.4. Economic features

Economic features are labelled E1 and E2 in Table 8. With regards to the cost of a tool, opinions differed (see Table 8 row E1). At the extremes, five participants thought free tools were Highly desirable whereas four participants thought free tools were not necessary.

Table 16 identifies the main discussion themes for economic features. The discussion of the cost of tools centred around the concern that it was not possible to get good quality, trustworthy tools that provided all required features without payment. Nine participants mentioned that they did not expect good tools to be free.

Three participants mentioned different licenses for different users would be a good idea, allowing free systems for students or for private use.

[4] I.e. a tool able to function, primarily, as a stand-alone application.

Table 16. Comments about economic features

ID	Feature	Theme	Participants	Experience
E1	Free	Good tools aren't free	9	L(4), M(1), H(4)
		Different licences for different users	3	L(1), M(1), H(1)
E2	Maintained	Methods evolve	4	L(2), M(1), H(1)
		Need defect management	2	L(1), H(1)

All participants felt post development maintenance of a tool (see Table 8 row E2) was either Mandatory (six participants) or Highly desirable (seven participants). The discussion of this feature concerned the need for maintenance, with four participants pointing out that methods evolve and two mentioning that such large, complex systems would probably include defects that would need to be corrected.

Overall trends

Several themes were identified against more than two features:

- Viability (i.e. the concern that the feature would be difficult to automate) was identified against seven different features.
- Time saving (i.e. the potential for a feature to substantially decrease the SR workload) was identified against five features.
- Use other tools (i.e. the availability of other tools to implement the feature requirements) was identified against three features. The specific features were Protocol Development, Reporting and Report Validation.
- For complex projects (i.e. the feature was considered appropriate for projects with large or distributed teams) was identified against three features. The specific features were Protocol Development, Multiple Users and Role Management.

Table 17 shows the number of times participants mention the issues of Viability and time saving for each SR process tool feature[5]. This table suggests that participants were most concerned about the viability of support for the search process, data extraction and reporting. In addition, participants identified time saving as likely for search automation, selection and data synthesis processes more often than for other processes.

Table 18 shows the distribution of comments concerning Viability and Time saving against individual participants. It shows the number of times each participant made a comment about each issue. The table shows that concerns about viability of tool support are spread across all but one of the participants. On the other hand, although only one participant with a high level of experience mentioned time saving four times, four out of six participants who mentioned time saving had low levels of experience suggesting the time taken to complete an SR is of more importance to relative novices. This is consistent with the results shown in Table 2, where five out of six participants who mentioned that SRs were generally time consuming had low levels of experience.

5.3.5. Comparison of importance ratings

Table 8 presents the assessment of the importance of the features to SE researchers. No assessment for the importance of reusing results from previous SRs was provided, because the reuse of past project data is seldom performed in SE systematic reviews, so there was possibility of rating the importance of this feature.

A comparison of the assessment results and the study participants' assessments shows that for every feature, the majority of participants agreed that it was important. Thus, the set of all features that should be included in a SRLC tool is quite robust to differences between domains. As it was expected, there were differences in the

[5]Time saving and Viability were not mentioned against any other feature groups.

Table 17. Distribution of general comments against features

Feature	Viability	Time saving
Protocol development	1	0
Protocol validation	0	0
Search process	5	3
Study selection	0	3
Quality assessment	2	0
Data extraction	4	0
Data synthesis	2	3
Text analysis	2	1
Meta-analysis	0	0
Reporting	4	1
Report validation	0	0

Table 18. Distribution of general comments against participants

Participant	Experience	Viability	Time saving
P01	M	1	1
P02	L	1	0
P03	L	5	1
P04	L	1	2
P05	M	1	0
P06	L	3	1
P07	L	2	1
P08	H	0	0
P09	H	2	0
P10	L	2	0
P11	L	2	0
P12	H	1	0
P13	H	1	4

evaluation of the importance of features among individual participants and among domains. However, there were also similarities.

For ten features, the modal response of participants to the importance of the feature was exactly the same as this assessment. In the case of three other features, there were two modal values for feature importance, and in both cases one of the modal values was the same as ours. In only three of the remaining features, did the modal value of the participants scores differ by more than one level from ours. The three features with substantial disagreement were:

1. Security, regarded as Desirable by the authors, had a modal value of Mandatory among the interview participants.
2. Meta-analysis, which we regarded as Nice-to-have, but which nine of the 13 interview participants rated as Mandatory or Highly desirable.
3. Role management, which was regarded as Highly desirable, while the modal response of the participants was Nice-to-have. However, it should also be noted that six of the participants rated this feature as Mandatory or Highly desirable.

These results confirm that the importance of various features is context dependent. For example, meta-analysis is rarely undertaken in SE research but is a normal part of health care research, so it is much less important to SE researchers than health care researchers. Nonetheless, although there are differences, it appears that the importance of features is surprisingly similar across the different domains. It should also be noted that none of the participants suggested any additional features which confirms that the SR methodology is not radically different in different domains.

5.4. The effect of experience on perceptions of feature importance

There has been considerable discussion in SE about the problems facing novice reviewers (see, for example, [12] and [11]). Furthermore, this issue was directly investigated by Hassler et al. [31]. Therefore the main interest was the investigation whether relative novices had different perceptions of the importance of tool features compared with more experienced reviewers.

Table 19 addresses exactly this issue. The column labelled *Total % Score* is the percentage of the maximum importance score obtained for a specific feature across all participants. The score was obtained by mapping the ordinal scale points for importance to numbers (i.e. Mandatory = 4, Highly desirable = 3, Desirable = 2, Nice to have = 1 and Not needed = 0). The total percentage importance score for a feature was obtained as follows:

$$TotalScore_i = 100\frac{\Sigma_j Importance_{i,j}}{\Sigma_j(4)} \qquad (1)$$

Table 19. Relationship between features scores and experience

ID	Feature	Total % Score	Low exp	High exp	Diff
SRM1	Multiple users	88.46	79.17	96.43	17.26
SRT6	Data extraction	86.54	79.17	92.86	13.69
E2	Maintained	86.54	75.00	96.43	21.43
SRT5	Quality assessment	82.69	79.17	85.71	6.55
SRT7	Data synthesis	82.69	70.83	92.86	22.02
SRT4	Study selection	80.77	70.83	89.29	18.45
IS1	Ease of Setup	80.77	70.83	89.29	18.45
SRM2	Document management	78.75	70.83	85.71	14.88
SRM5	Reuse of past data	75.00	66.67	82.14	15.48
SRT9	Meta-analysis	71.15	58.33	82.14	23.81
IS2	Installation guide	69.12	58.33	78.57	20.24
IS3	Tutorial	69.12	58.33	78.57	20.24
SRM3	Security	67.31	50.00	82.14	32.14
SRT3	Search process	65.38	79.17	53.57	-25.60
IS4	Self-contained	57.69	54.17	60.71	6.55
SRM4	Role management	55.77	33.33	75.00	41.67
SRT1	Protocol development	51.92	50.00	53.57	3.57
SRT10	Reporting	46.15	45.83	46.43	0.60
E1	Free	42.31	37.50	46.43	8.93
SRT2	Protocol validation	34.62	37.50	32.14	-5.36
SRT8	Text analysis	34.62	29.17	39.29	10.12
SRT11	Report validation	34.62	37.50	32.14	-5.36

where $TotalScore_i$ is the percentage of the maximum score for feature i, and the maximum score for a feature is $\Sigma_j(4)$, $j = 1, \ldots, 13$ is the number of participants and $Importance_{i,j}$ is the importance score that participant j gave to feature i. The table is ordered on this column.

The column labelled Low exp reports the percentage score for the six participants who had performed between one and five SRs and the column labelled High exp reports the percentage score for the seven participants who had completed more than five SRs. The column labelled Diff is the difference between the High exp score and the Low exp score.

Table 19 shows that, in general, participants with high levels of experience rated tool features higher than relative novices, since only three of the 22 features were scored higher by the relative novices than by the experienced participants.

It also seems that the relative importance of tools is quite similar for both groups, since the Pearson correlation between the scores for relative novices and experienced staff was 0.76. There are three features which exhibit extremely anomalous values:

1. Search process support was scored much lower by experienced participants than by relative novices.
2. Role management support was scored much higher by experienced participants than by relative novices.
3. Security support was also scored much higher by experienced participants than by relative novices but is not such an extreme anomaly.

Excluding these feature increases the correlation between the scores to 0.95.

5.5. The effect of SR type and domain

The authors hoped to assess whether the type of systematic review researchers performed influenced their perception of the importance of different framework features. For example, the authors expected researchers who primarily undertook quantitative systematic reviews to emphasise the importance of meta-analysis tools and researchers who primarily undertook qualitative systematic reviews to emphasise the importance of more general data synthesis facilities and text analysis facilities. It was also expected that social science researchers would undertake

Table 20. Experience and importance scores for analysis features

Experience	SR type	Domain	Meta-analysis	Data synthesis	Text analysis
Low	Quant	HC	3	3	2
Low	Qual	HC	3	2	3
Low	Qual	HC	1	3	0
Low	Quant	HC	2	3	0
Low	Quant	SS	4	3	1
Low	Both	SS	1	3	1
High	Both	HC	4	4	0
High	Both	SS	4	4	3
High	Quant	HC	3	4	1
High	Both	SS	3	4	2
High	Both	SS	2	4	3
High	Qual	SS	3	3	1
High	Both	HC	4	3	1

Table 21. The impact of domain and SR type on scores for analysis features

Factor	Type	Participants	Meta-analysis	Data synthesis	Text analysis
Domain	HC	7	71.43	78.57	25.00
	SS	6	71.43	71.57	35.71
SR type	Both	6	75.00	91.67	41.67
	Qual	3	58.33	66.67	33.33
	Quant	4	75.00	81.25	25.00

qualitative systematic reviews and health care researchers would undertake primarily quantitative systematic reviews.

The expectations of the authors were not met. Table 20 shows the systematic review type, Domain type of participants and their importance scores for meta-analysis, data synthesis and text analysis. Four of the social science participants and two from health care reported performing both quantitative and qualitative systematic reviews. Of the remaining five health care researchers, three concentrated on quantitative systematic reviews and two on qualitative systematic reviews. Of the remaining two social sciences participants, one primarily undertook qualitative studies and the other primarily undertook quantitative studies. The impact of the domain and SR type are summarized in Table 21. In the case of tool support for meta-analysis and data synthesis, Table 19 shows that more experienced participants tended to regard such a feature to be more important than the less experienced ones, however, Table 21 suggests that there is no domain effect.

With respect to SR type, Table 21 suggests that participants doing qualitative studies may regard support for meta-analysis and data synthesis as less important than other subjects. However, this result may be confounded with experience since only two of the seven subjects who concentrated on a single study type had high levels of experience whereas five of the six subjects who did both types of study had high levels of experience.

5.6. Revising the setup and installation features

During the previous validation of the SRLC tool framework [18] it was difficult to distinguish between the three features related to installing and using the SRLC tool and therewas an idea that would be better to integrate the three features into a single feature. The scores given by each

Table 22. Experience and importance scores for features related to installation and set up

Experience	Ease of set up	Installation guide	Tutorial
Low	4	1	2
Low	3	3	2
Low	3	3	3
Low	3	3	3
Low	1	1	1
Low	3	3	3
High	4	1	1
High	4	4	4
High	4	4	4
High	4	4	4
High	2	2	2
High	3	3	3
High	4	4	4

participant to each of the three features is shown in Table 22. Across the three features, 10 of the 13 participants gave the same score for all three features. Those that gave different scores, scored the Installation guide and Tutorial lower than Ease of Set up. This result supports the view that only one high-level feature is needed to address the set up and installation.

However, participants' earlier comments relating to the difficulty of using EPPI-reviewer and RevMan (see Table 7) suggest that usability is a significant issue to users. Therefore, a feature relating to provision of a Tutorial should be included. However, it might be preferable to generalise the feature and use the term *Ease of Use*, with a tutorial as one way of implementing such a feature.

6. Discussion

In this section the results of this cross-domain study is discussed from the viewpoint of the research goals.

6.1. The relevance of experiences from other domains

The results show that there are some differences between SE reviews and those in health care and social sciences. For example, health care and social science researchers may undertake systematic reviews commissioned by clients, whereas in SE these are normally researchers that undertake systematic reviews to further their own research goals.

There were other differences which the authors believe are likely to be due to the relative immaturity of systematic reviews in software engineering. For example, in Hassler et al.'s study [31] researchers with a high level of experience were defined as those who had performed three or more SRs, whereas in this study the highest experience levels of more than 15 SRs were categorized. In addition, reports from SE researchers summarized in [9] concentrated on technical processes which were emphasized in the first two versions of the SE systematic review guidelines. In contrast comments from the participants of this study identified issues related to review management not only issues related to technical processes. This is consistent with the results of Hassler et al.'s study [31] in which he noted that researchers with higher experience levels voted for features that aided tactical activities, whereas novices voted mainly for tools supporting operational tasks. As researchers in software engineering begin to perform more complicated systematic reviews, both in terms of SRs that involve many distributed researchers, as well as studies that involve large numbers of candidate primary studies, possibly of different study types, it was expected that SE researchers would experience more problems associated with systematic review management.

Another difference was that there were two additional challenges mentioned by study participants that were not considered in the SE literature: version control and formatting references. Both of these issues seem important in a comprehensive SRLC tool, so need to be considered in any comprehensive evaluation framework for SRLC tools.

We also observed some differences in the ratings of importance of SLRC tool features, compared with our assessment of the importance of such features to SE researchers:

- Support for meta-analysis appeared to be more important to participants than it was assessed to be to SE researchers in this study. This was true even for two of the three participants who primarily undertook qualitative reviews. It appeared that study participants were well aware that meta-analysis tools are essential for some quantitative studies, even if they did not use such tools themselves.
- Support for security was more important in the health care and social science domains than it is in SE. In particular, more experienced participants were very concerned about restricting access to confidential information (only one of the five participants who mentioned this was a relative novice), whereas two relative novices felt that since they were dealing with existing published papers confidentiality was not an issue. In terms of SE researchers, it would certainly be the case that mapping studies were unlikely to have any confidentiality issues.
- There was a lack of strong support for textual analysis tools. Kitchenham and Brereton [9] reported that there were a substantial number of software engineering studies addressing textual analysis for systematic reviews and Marshall and Brereton [9] identified the number of tools to support textual analysis, so more enthusiasm was expected for such a feature. However, the modal response among the 13 participants was that such a feature was only "Nice-to-have". Nonetheless, participants were enthusiastic about other features that could be implemented using textual analysis such as study selection (modal response "Highly desirable") and data synthesis (modal response "Highly desirable") and one user of EPPI-reviewer pointed out that EPPI-reviewer used textual analysis to implement a feature that finds the most relevant studies. It was concluded that textual analysis may be necessary in order to implement SRLC tool features, but it may not be needed as a top level feature available directly to tool users.

Overall, it was concluded that there are common challenges among the different domains and the results of this study could be used to to evaluate and refine our evaluation framework. Furthermore, since the domains have similar challenges, it is in the interest of software engineering researchers to remain aware of innovations in the systematic review methodology to avoid the risks of both missing out on new methods or re-inventing the wheel.

6.2. Tools used to support systematic reviews

Participants identified 14 tools that they used while doing systematic reviews. The most commonly used tools were reference manager tools in particular RefWorks and EndNote. In addition, the participants mentioned two SRLC tools: RevMan and EPPI-reviewer. However some of the tools were general purpose tools such as Microsoft Word and Excel, while others were statistical software tools or bespoke tools. The core set of ten tools that support systematic reviews including reference managers, SRLC tools and meta-analysis tools, together with tools identified in Marshall and Brereton's mapping study [21] and tools identified from other sources (i.e. [28, 34], and the Cochrane Collaboration website) were incorporated into an online tool called SRToolbox [29]. This set of tools has been substantially updated since this research was completed, and the most up-to-date categorized list can be found at the website systematicreviewtools.com. This website is maintained by Marshall and has replaced the Cochrane Collaboration web pages on tools.

With respect to SRLC tools, EPPI-reviewer was believed to be relevant to the needs of SE researchers, however, it is unclear to what extent it is tailored specifically to the needs of social scientists, and it is not free.

In the context of features required in SRLC tools, a common discussion point with our participants was whether it was even possible to automate some of the features. Participants of all experience levels feared that advanced tools might be untrustworthy, in particular that they would miss things or make classification errors or be incomplete. Thus, SRLC tool developers need to have a sound rationale for the algorithms they use to implement features, before their tools are likely to be widely accepted. Furthermore potential tool users in SE should appreciate the difficulty of implementing some of the features they might desire.

Another important issue was that most participants did not expect good quality tools to be free. Also the participants agreed that tools needed to be maintained because methods evolve and complex tools usually have residual errors that need to be corrected.

6.3. The impact of participant experience

Generally, more experienced participants rated features of support tools as more important than relatively inexperienced participants. It is likely that the more experienced participants had taken part in some large, complex systematic reviews and have, therefore, experienced the problems that such reviews can cause. Certainly, there is some evidence that more experienced participants undertook more varied SRs. Table 1 shows that five of the six relative novices undertook only one type of SR (either qualitative or quantitative) whereas only two of the seven more experienced researchers performed only one type of study.

The implication for SE researchers is that the need for SE tools in general, and SRLC tools in particular, should be expected to increase as SE researchers become more experienced with the SR process, and attempt larger and more complex systematic reviews. In particular, Table 2 and Table 8 indicate the importance of tools to support SR process management in addition to tools supporting specific SR tasks.

Throughout this study, the participants often mentioned that the importance of tool features depended on the size of the team and the complexity of the SR. Thus, requirements for SRLC tools should probably be elicited from researchers who have experienced the problems of large-scale SRs. In addition, the evaluations of such tools should ideally involve experienced researchers and large-scale SRs.

Also, since novice researchers usually undertake relatively small reviews in small teams, they might be best served by using a variety of tools, including Microsoft Excel and Word and a reference manager system, that they are already familiar with. It is unlikely that novices would benefit from extensive automation if the overheads, such as the required learning time needed to use a tool effectively, are significant.

6.4. Implications for the evaluation framework

One of the main aims of the study was to provide some independent assessment of the SRLC tool evaluation framework [18]. Kitchenham and Brereton had been deeply involved in the adoption of systematic reviews in SE. Originally, the promoted process was developed from the health care domain and the main focus was on on adapting the methodology to the SE domain. After developing the evaluation framework based on SE practice, it was thought that it would be extremely valuable to investigate whether there were more insights to be obtained from other domains.

The discussion about the features of an SRLC tool and the relative importance of such features confirmed that all of the features and the majority of the importance ratings were consistent with the views of the health care and social science researchers. In particular, none of the features was considered completely unnecessary and only three features had importance ratings very different from the ratings obtained in this study.

However, some changes were made in the evaluation framework as a consequence of the study results:

1. Analysis of the three features related to the ease of installation and setup confirmed the view that it was better to have only one feature labelled *Ease of Setup*, where installation guides are a means by which the feature can be implemented. In addition, since several participants commented that RevMan and EPPI-reviewer were difficult to use, it was recommended to replace the Tutorial feature by the *Ease of use* feature, with a tutorial as one means of assisting tool users to use the tool effectively. The feature set should be renamed as *Usability*.
2. The discussion about the importance of textual analysis convinced us that it was not really a self-standing feature, but represented a means of supporting various features such as *Data synthesis* and *Study selection*. The evaluation framework includes additional assessment criteria to assist evaluating how well each feature is implemented. Now the textual analysis is included as one of the additional criteria used to assess the support for these features.
3. Three challenges that were mentioned by participants but had not been discussed in the SE literature were identified. One of them was *negotiating with policy makers* which does not appear to be an issue of relevance to software engineering researchers, and indeed, may only be of relevance in the UK to health care and social science researchers. The other two issues were *version control* and *formatting references*. Both of these issues should be of concern to software engineering researchers. Version control was already mentioned in the evaluation framework as an associated assessment criteria for the protocol development but it should also be included in the associated evaluation criteria for report development. Formatting references should be included in the additional assessment criteria of the Search process.
4. Importance level was not assigned to the *Reuse of Past Project Data*. It was decided to adopt the rating of Highly desirable which was the modal value of the participants' ratings. However, the users of this evaluation framework are expected to downgrade the importance level if they do not plan to keep their SR results up to date.

The changes have only a limited effect on the evaluation framework. For example, the SLuRp tool [20] would have scored 65% with the framework as it was used before this study. The tool score is the weighted sum of the score for each feature set: where the weight for the SR activity feature set is 4, the weight for the Process Management feature set is 3, the weight for the Usability feature set is 2, and the weight for the Economic feature set is 1:

$$ToolScore = \frac{\Sigma_{i=1,...,4} FSW_i FSS_i}{\Sigma_{i=1,...,4} FSW_i} \qquad (2)$$

where FSS_i is the score for feature set i, and FSW_i is the weight for feature set i.

The score for each feature set is the sum of the score for the extent to which each feature is supported (taking values 0, 0.5 and 1) multiplied by the score of the importance of each feature. This value is converted to the percentage of the maximum score for the feature set:

$$FSS_i = \frac{100\Sigma_{j=1,...,k} FI_j FS_j}{\Sigma_{j=1,...,k} FI_j} \qquad (3)$$

where FSS_i is score for feature set i, FI_j is the numerical importance for feature j in feature set i and FS_j is the extent to which the feature is supported in the tool being evaluated.

As a result of the changes introduced by this study the score for SLuRp decreased to 63% because:

- The feature Ease of Setup was scored as partly true for SLuRp and was given an implementation value of 0.5, since an installation guide was available.
- The feature Installation guide was removed as a separate feature in the framework decreasing the number of features in the Usability feature set to four.
- The feature Ease of use was introduced as a feature (to replace the Tutorial feature) with an importance of Highly desirable.

SLuRp scored the minimum value of zero for the feature since there was no tutorial, nor an online help facility, and the system is very complex.
- The feature Re-use of past data was included in the Process Management feature set, with an importance level of Highly desirable. SLuRp maintains records of past SRs and their results, so it scored the maximum value of one for this feature.
- Text analysis on which SLuRp scored the maximum value of one was removed as a feature in the SR activity support feature set.

6.5. Comparison with other results

As reported in Section 3, Hassler and his colleagues undertook a series of studies investigating SR tool requirements. In contrast to the results reported in this study, their studies concentrated on the opinions and experiences of the SE community.

Carver et al. [14] investigated barriers to the SR process. Many of the issues they mentioned were discussed in Kitchenham and Brereton's systematic review [9]. However, they also provided a much more detailed discussion of the problems with current SE databases including the necessity to deal with duplicates, which was mentioned by one of the participants in this study. They also mentioned the issue of coordinating the reviewing and selection of papers and associated issues for team management and conflict resolution which were mentioned by the participants of this study.

The participants in Carver et al.'s study ranked the SR processes as most in need of tool support. They ranked Searching Databases as most important followed by Selecting papers and Extracting data. In contrast, this study rated Data Extraction as the most important SR task requiring support, followed by Quality Assessment and Data Synthesis. This difference may be caused by the concentration on mapping studies in SE. Carver et al.'s results suggested relatively little support for issues related to protocol development (i.e. Defining Research Question, Identifying Keywords, and Creating Search Strings), which is consistent with the relatively low importance given by our participants to automated support for protocol development.

It is quite difficult to make detailed comparisons between Hassler et al.'s study to identify barriers to the SR process [30] and [31] this one, because in each study, the terminology was based on the terminology used by the participants. In addition, when the participants of Hassler's studies voted, their votes were constrained. They were given a number of tokens (i.e. votes) and these tokens were shared across all the features being voted on and participants could give multiple tokens to specific features. This process meant that participants were prioritising across all the possible tools. In this study the participants were not asked to make any trade-off when they assessed the importance of individual tool features.

Hassler et al. [30] identified barriers faced by systematic reviewers related to the SR process, primary studies, the practitioner community and tooling. The comparison of the discussion points in Hassler's study with the results of this study is shown in Table 23. Hassler identified the difficulty of meta-analysis as a problem, but looking at his comments it appears that data synthesis rather than statistical meta-analysis was a problem, which is consistent with these results. Barriers related to the practitioner community were not mentioned as a problem in health care or social science where the practitioner community may be more accustomed to the need for systematic reviews. Hassler's participants identified barriers related to tooling in terms of needing improved search and retrieval facilities including addressing the problem of rewriting search engine strings which was mentioned as a challenge by one participant. However, support for the search process did not feature as one of the most important features in Table 19. It is noticeable that support for the search process is considered much more important by relative novices than by experienced researchers, so the difference between our result and Hassler's results may reflect the fact that there are few researchers in SE that have completed more than 5 systematic reviews. Hassler discussed the need for support for data extraction and management. Our results strongly align with this result, since support for Data Extraction was the

Table 23. Comparison with barriers discussed in Hassler's study [30]

Category	Issue	This study
SR process	SR protocol is sequential, but process iterative	Not mentioned
	Meta-analysis is difficult	Need support for data synthesis
	Lack of methods for result interpretation	Not mentioned
Primary studies	Title and abstracts misleading	Not mentioned
	Terminology not standardized	Mentioned by four participants
Practitioner community	Difficulty relating to industry needs	Not mentioned
	Difficulty justifying structured process	Not mentioned
Tooling	Electronic databases are inadequate for search and retrieval	Problem with string translation mentioned once
	Need data extraction and management tools	Strong support in this study

second most highly ranked feature by our participants and features related to Management issues, such as Multiple Users, Document Management, Role Management which were all highly ranked particularly by more experienced researchers.

Hassler et al. undertook a second community workshop to identify SR tool needs [31]. In this workshop they had 16 participants of which 10 were categorized as "experts" because they had completed at least three SRs. They compared their results with those of Marshall et al. [19] In this study this analysis was extended to consider the impact of participant experience as shown in Table 24. This table is ordered on the total score for the features in this study. The order of the total score for equivalent features in Hassler's study is shown in parenthesis after the name of the feature. The experience scores for high and low experience participants were included, however, it is important to note that high experience was equated with completing more than five SRs so the comparisons are not exact. One change was introduced to Hassler et al.'s table, that is the Textual analysis feature was equated to Hassler's Automated Analysis rather than to Statistical Analysis.

The most obvious area of agreement between the study results is that, given that Multiple Users and Collaboration are equivalent, they correspond to the most important feature in this study and the second most important in Hassler's study, with importance rated more highly by more experienced researchers.

However, there are major differences between the ranking of tool features. The correlation between the total scores for this study and for Hassler et al.'s study is 0.44. Furthermore, the correlation between the scores for participants with low experience was 0.24, and between scores for high experience participants was 0.25. In addition, the correlation between the high and low experience participants' votes in Hassler's study was only 0.45.

Differences between Hassler's results and the ones obtained in this study could be due to the specific participants but it could also be caused by domain differences, experience differences or differences in the type of SRs in the SE domain. It is suspected that a major issue is the difference resulting from the prevalence of mapping studies in SE. Mapping studies are often confused with SRs in the SE community. However, they are often published in conferences and journals implying that mapping studies are of value to the SE community. This is not the case in health care or social sciences. Concentrating on mapping studies can lead to SE researchers being more interested in the search and selection processes than researchers in other domains and less concerned about data extraction and quality assessment. Also a mapping study analysis is often concerned with the similarities between large numbers of studies which is helped by visual analysis and textual analysis techniques. Thus the relevance of results from other domains may depend on the extent to which systematic review

Table 24. Comparison of features scores and experience for this study and Hassler et al.'s study [31]

Our study				Hassler et al.			
Feature	Total	Low exp	High exp	Feature	Total	Low exp	High exp
Multiple users	88.46	79.17	96.43	Collaboration (2)	10.7	4.9	13.3
Data extraction	86.54	79.17	92.86	Coding (= 8)	3.8	4.9	3.3
Quality assessment	82.69	79.17	85.71	Quality assessment (= 5)	5.3	2.4	6.7
Data synthesis	82.69	70.83	92.86	Automated analysis (= 5)	5.3	9.8	3.3
Study selection	80.77	70.83	89.29	Study selection (3)	6.9	9.8	5.6
Document management	78.75	70.83	85.71	Study storage (= 8)	3.8	2.4	4.4
Reuse of past data	75.00	66.67	82.14	Data maintenance (4)	6.1	7.3	5.6
Meta-analysis	71.15	58.33	82.14	Statistical analysis (11)	2.3	4.9	1.1
Search process	65.38	79.17	53.57	Integrated search (1)	11.5	9.8	12.2
Protocol development	51.92	50.00	53.57	Development & validation (= 12)	0.8	0.0	1.1
Reporting	46.15	45.83	46.43	NA			
Protocol validation	34.62	37.50	32.14	Development & validation (12)	0.8	0.0	1.1
Text analysis	34.62	29.17	39.29	Automated analysis (= 5)	5.3	9.8	3.3
Report validation	34.62	37.50	32.14	Report validation (10)	3.1	2.4	4.4

approaches in SE continue to be dominated by mapping studies.

Some differences may be caused by the relatively low levels of experience among SE researchers. The high and low experience participants in Hassler's study are probably closer to the low experience participants in our study. So the differences between high and low studies in Hassler's study are more likely to be chance effects than those in this study.

6.6. Limitations

A major limitation of this cross-domain study is that the use of systematic reviews was discussed, however, mapping studies (or scoping studies as they are often referred to in other domains) were not explicitly discussed. Although the participants did not raise the issue of such studies themselves, it is possible that the assessment of the importance of some SRLC tool features might have changed if we had asked them to consider the implications of the features for scoping studies. A particular issue for software engineering SRLC tools is that textual analysis may well play a more important role in managing the study selection and data extraction for mapping studies than it does for systematic reviews. However, we would still expect textual analysis to be used to implement various features rather than being a tool feature in its own right.

Another important limitation is that there were relatively few participants. Nonetheless, the coverage of the three characteristics thought to have some influence on participants' experience was good: domain, type of SRs they undertake, and their level of experience. This means that the group of participants was heterogeneous, which is often considered the best approach to obtain a theoretical sample for a qualitative study.

All of the study participants were UK-based, so this might introduce some cultural bias into the study. However, all versions of the SE systematic review guidelines were based primarily on UK standards and they were widely adopted among software engineers from many different countries. Thus, our SR practices in software engineering may already have a built-in UK cultural bias.

Yet another limitation of this cross-domain study are those related to the method of semi-structured interviews and the experience of the interviewer. Since this study was part of Marshall's PhD research, he performed all the reviews himself. However, in general, interview-based studies might be improved by the use of observer

triangulation. In addition, semi-structured interviews depend strongly on the communication skills of the interviewer [35]. Marshall attempted to address this issue by undertaking a pilot study. Other risks are associated with the participants' impression of the interviewer. Research suggests that people respond differently depending on how they perceive the interviewer (*the interviewer effect* [36]). Factors such as gender, age and the ethnic origins of the interviewer have a bearing on the amount of information people are willing to contribute [36]. In addition, participants' responses can be influenced by what they think the situation requires [37]. Marshall did all the interviews and made every effort to put the participants at ease and to explain the purpose of the interview. In addition, the fact that he was reasonably knowledgeable about systematic reviews and systematic review tools was found useful in overcoming potential problems due to his relatively junior level. Risks associated to missing relevant questions as the participants lead the flow of the interview were mitigated by using a list of questions and key themes to check the progress of the interview.

7. Conclusions

The results of our cross-domain study suggest that, in the context of systematic reviews, experiences of researchers in other disciplines can be valuable for SE researchers. The implications of this are:

- Standalone tools used by systematic reviewers in other domains may be of value to systematic reviewers in SE. We recommend SE researchers, particularly those supervising junior researchers, to periodically consult the SR Toolbox to keep track of available tools.
- SE researchers producing tools for systematic reviews should also be aware of the currently available tools and their features. In particular, in the context of SRLC tools, the features available in the EPPI-reviewer tool might be worth studying.
- SE researchers can benefit from keeping abreast of systematic review developments in other disciplines. This is important to avoid a methodological drift. Researchers should not want general scientific methods to start to diverge across different domains. Nonetheless, there are some differences between domains that can impact the adoption of standards or tools, such as the importance of mapping studies, which makes it useful for SE researchers to continue to study SR methodology.

In terms of the impact of the results reported in this paper, we made several changes to our framework for evaluating SRLC tools. The changes were easy to implement and overall it appeared that the framework was quite robust across different domains [38].

We intend to continue refining the evaluation framework's feature set and evaluation criteria to accommodate the selection and assessment of novel tools developed to support systematic reviews. For example, a case study is currently under way to compare and evaluate a selection of tools that support network meta-analysis which uses an expanded version of the evaluation framework. Further refinements to the framework will also be reflected as part of the ongoing development of the Systematic Review Toolbox to classify tools.

Acknowledgements

We would like to thank the participants in the study for sharing their experiences with us and the reviewers for their helpful comments on our manuscript.

References

[1] C. Mulrow, "Rationale for systematic reviews," *British Medical Journal*, Vol. 309, No. 6954, 1994, p. 597.

[2] D. Cook, C. Mulrow, and R. Hayes, "Systematic reviews: synthesis of best evidence for clinical decisions," *Annals of Internal Medicine*, Vol. 126, No. 5, 1997, pp. 376–380.

[3] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in soft-

ware engineering," Keele University and Durham University, Joint Report, 2007.

[4] D.S.W. Rosenberg, J. Gray, R. Hayes, and W. Richardson, "Evidence-based medicine: what it is and what it isn't," *British Medical Journal*, Vol. 312, No. 7023, 1996, p. 71.

[5] J. Higgins, *Cochrane Handbook for Systematic Reviews of Interventions*. Wiley-Blackwell, 2008.

[6] J.P. Ioannidis, "The mass production of redundant, misleading and conflicted systematic reviews and meta-analysis," *The Milbank Quarterly*, Vol. 94, No. 3, 2016, pp. 485–514.

[7] D.S. Cruzes and T. Dybå, "Research synthesis in software engineering: A tertiary study," *Information and Software Technology*, Vol. 53, No. 5, 2011, pp. 440–455.

[8] F.Q. da Silva, A.L. Santos, S. Soares, A.C.C. França, C.V. Monteiro, and F.F. Maciel, "Six years of systematic literature reviews in software engineering: An updated tertiary study," *Information and Software Technology*, Vol. 53, No. 9, 2011, pp. 899–913.

[9] B. Kitchenham and P. Brereton, "A systematic review of systematic review process research in software engineering," *Information and Software Technology*, Vol. 55, No. 12, 2013, pp. 2049–2075.

[10] P. Brereton, B.A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of Systems and Software*, Vol. 80, No. 4, 2007, pp. 571–583.

[11] M. Babar and H. Zhang, "Systematic literature reviews in software engineering: preliminary results from interview with researchers," 2014, pp. 346–355.

[12] M. Riaz, M. Sulayman, N. Salled, and E. Mendes, "Experiences conducting systematic reviews from novices' perspective," in *Proceedings of the 2010 International Conference on Evaluation and Assessment in Software Engineering*, 2010.

[13] S. Imitiaz, M. Bano, N. Ikram, and M. Niazi, "A tertiary study: Experiences of conducting systematic literature reviews in software engineering," in *In Proceedings of the 2013 International Conference on Evaluation and Assessment in Software Engineering*, 2013, pp. 177–182.

[14] J. Carver, E. Hassler, E. Hernandes, and N. Kraft, "Identifying barriers to the systematic literature review process," in *Proceedings of the 13th International Symposium on Empirical Software Engineering and Measurement*, 2013.

[15] M. Staples and M. Niazi, "Experience using systematic review guidelines," *Journal of Systems and Software*, Vol. 80, No. 9, 2007, pp. 1425–1437.

[16] H. Ramampiaro, D. Cruzes, R. Conradi, and M. Mendona, "Supporting evidence-based software engineering with collaborative information retreival," in *Proceedings of the 2010 International Conference on Collective Computing: Networking Applications and WorkSharing*, 2010, pp. 1–5.

[17] B.A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-Based Software Engineering and Systematic Reviews*. CRC Press, 2015.

[18] C. Marshall, O.P. Brereton, and B.A. Kitchenham, "Tools to support systematic literature reviews in software engineering: A feature analysis," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE'14)*. ACM Press, 2014, pp. 13:1–13:10.

[19] C. Marshall, O.P. Brereton, and B.A. Kitchenham, "Tools to support systematic literature reviews in software engineering: A cross-domain survey using structured interviews," in *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering (EASE'15)*. ACM Press, 2015, pp. 26–31.

[20] D. Bowes, T. Hall, and S. Beecham, "SLuRp: A tools to help large complex systematic literature reviews," in *Proceedings of the 2012 International Workshop on Evidential Assessment of Software Technologies*, 2012, pp. 33–36.

[21] C. Marshall and P. Brereton, "Tools to support systematic literature reviews in software engineering: A mapping study," in *Proceedings of ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE Computer Society Press, 2013, pp. 296–299.

[22] J. Molleri and F. Benitti, "SEAR: A web-based automated tool to support the systematic literature review process," in *Proceedings of the 2015 International Conference on Evaluation and Assessment*, 2015, pp. 24–33.

[23] B. Kitchenham, "Evaluating methods and tool Part 1: The evaluation context and methods," *ACM SIGSOFT Notes*, Vol. 21, No. 1, 1996, pp. 11–14.

[24] B. Kitchenham, T. Dybå, and M. Jørgensen, "Evidence-based software engineering," in *Proceedings of ICSE 2004*. IEEE Computer Society Press, 2004, pp. 273–281.

[25] B. Kitchenham, "Procedures for undertaking systematic reviews," Keele and Durham Universities, Joint Technical Report, 2004.

[26] M. Petticrew and H. Roberts, *Systematic Reviews in the Social Sciences: A Practical Guide*. Blackwell Publishing, 2006.

[27] J.A.S. Torres, D.S. Cruzes, and L. Salvador, "Automatic results identification in software engineering papers. Is it possible?" in *Proceedings of the 12th International Conference on Computer Science and Its Applications*, 2012.

[28] K.R. Felizardo, S. MacDonell, E. Mendes, and J. Maldonado, "A systematic mapping on the use of visual data mining to support the conduct of systematic literature reviews," *Journal of Systems and Software*, Vol. 7, No. 2, 2012, pp. 450–461.

[29] C. Marshall and O.P. Brereton, "Systematic review toolbox: a catalogue of tools to support systematic review," in *Proceedings of 19th International Conference on Evaluation and Assessment in Software Engineering (EASE'15)*. ACM Press, 2015, pp. 26–31.

[30] E. Hassler, J.C. Carver, N.A. Kraft, and D. Hale, "Outcomes of a community workshop to identify and rank barriers to the systematic literatire review process," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, 2014.

[31] E. Hassler, J.C. Carver, D. Hale, and A. Al-Zubidyb, "Identification of SLR tool needs–results of a community workshop," *Information and Software Technology*, Vol. 70, 2016, pp. 122–129.

[32] D. Remenyi, *Grounded Theory: A reader for Researchers, Student, Faculty and Others*, 2nd ed. Academic Conferences and Publishing International Limited, 2014.

[33] M.B. Miles, A.M. Huberman, and J. Saldaña, *Qualitative Data Analysis: A Methods Sourcebook*, 3rd ed. Sage Publications Inc., 2014.

[34] G. Tsafnat, P. Glasziou, M. Choong, A. Dunn, F. Galgani, and E. Coiera, "Systematic review automation technologies," *Systematic Reviews*, Vol. 3, No. 1, 2014, p. 74.

[35] P. Clough and C. Nutbrown, *A student's guide to methodology*, 3rd ed. SAGE, 2012.

[36] M. Denscombe, *The good research guide: for small-scale social research*. Open University Press, 2014.

[37] R. Gomm, *Social Research methodology*. Palgrave MacMillan, 2004.

[38] C. Marshall, *Tool support for systematic reviews in software engineering*, Ph.D. dissertation, School of Computer Science and Mathematics, Keele University, 2016.

Appendix A. Interview guide

The interview guide was intended to help structure the interview and ensure that all relevant points were covered. Since these interviews were semi-structured, it might be the case that not all questions were required. Similarly, supplementary questions, not recorded in this guide, could be asked, depending on the individual circumstances of each interview. Questions have been classified into four groups; namely, Group 1: Subject Context, Group 2: Personal Experience with Systematic Reviews, Group 3: Experience with Tools to Support Systematic Reviews and Group 4: Features for a Tool to Support Systematic Reviews.

A1. Introduction

Welcome the participant and ensure they are suitably comfortable, etc. Explain the purpose of the interview again so as to gather information about tools to support systematic reviews.

A2. Group 1: G1 subject context

Questions in Group 1 will be asked about the participants' discipline. In particular, we are interested in discovering how SRs are used within the domain, the infrastructure provided when undertaking a SR and any tools that are available to support the process. Four questions will be asked.

G1-Q01. Could you tell me about your discipline?

G1-Q02. How do systematic reviews play a role within your discipline?

G1-Q03. What infrastructure does your discipline provide to support reviewers when performing an SR? (e.g. guidelines)

G1-Q04. What tools to support SRs are available within your discipline?

A3. Group 2: G2 personal experience with systematic reviews

Questions in Group 2 will be asked about the participants' personal experience when performing an SR. In particular, we are interested to learn the extent of their experience, their thoughts on the usefulness of SRs, what they believe to be the main challenges and which aspects they feel are most in need of support.

G2-Q01. How many SRs have you performed?

G2-Q02. Do you find SRs useful?

G2-Q03. What, in your opinion, are the main challenges when undertaking a SR?

G2-Q04. In your experience, what are the key aspects of the SR process that you feel are most in need of automated tool support?

A4. Group 3: G3 experience with tools to support systematic reviews

The questions asked in Group 3 will depend on whether or not the participant has experience using a tool to support them whilst undertaking an SR. If the experience exists, the participant will be asked about their experience using the tool(s). If the participant has not used a tool before, they will be asked why they haven't and whether they might consider using one in the future. In addition, question G3-Q09 initiates the snowballing sampling technique.

G3-Q01. Generally, do you feel the SR process could benefit from automated support?

G3-Q02. Have you used a tool (or multiple tools) to support yourself whilst undertaking a SR?

If the participant has experience using a tool, ask questions G3-Q003 to G3-Q06. If they have no experience using a tool, advance to question G3-Q07.

G3-Q03. What is the tool called? (This might have already been identified by question G1-Q04.)

G3-Q04. In your opinion, what were the main strengths of the tool?

G3-Q05. What were its key weaknesses?

G3-Q06. Overall, did you feel that using the tool was useful? (i.e. did you feel sufficiently supported?)

G3-Q07. Would you use the tool again?

G3-Q08. Is there a particular reason why you haven't used one? (e.g. don't know enough about them, don't feel they are necessary, etc.)

G3-Q09. Would you consider using one in the future?

G3-Q10. Do you know someone who has used one? (Snowball sampling.)

A5. Group 4: G4 features for a tool to support systematic reviews

Questions in Group 4 involve a data collection exercise. The interviewer will explain that a set of features for a tool to support the overall SR process has been developed. In their opinion, and in the context of the SR process within their discipline, the participant will be asked to determine whether each feature is considered "Mandatory (M)", "Highly desirable (HD)", "Desirable (D)" or "Nice-to-have (N)". Alternatively, the participant can decide that a feature is "Not necessary (NN)". The interviewer will record the ratings made by each participant using a form with a row for each feature and a column for each rating level.

A5.1. Feature Set 1 (F1): economic G4-F1

Questions relating to this feature set concern economic factors relating to the initial cost of the tool and the subsequent support for maintaining (or upgrading) the tool. Three questions will be asked.

G4-F1-Q01. How important is it that a tool should not require financial payment to be used?

G4-F1-Q02. How important is a well and freely maintained tool?

G4-F1-Q03. Are there any features you can think of that you might add to this feature set?

A5.2. Feature Set 2 (F2): ease of introduction and setup G4-F2

Questions relating to this feature set focus on the level of difficulty inherent in setting up and using the tool for the first time. Five questions will be asked.

G4-F2-Q01. How important is a simple installation and setup procedure?

G4-F2-Q02. How important is the presence of an installation guide?

G4-F2-Q03. How important is the presence of a tutorial?

G4-F2-Q04. How important is it that the tool is as self-contained as possible? (i.e. able to function as a stand-alone application with minimal requirements from other external technologies.)

G4-F2-Q05. Are there any features you can think of that you might add to this feature set?

A5.3. Feature Set 3 (F3): SR activity support G4-F3

Questions relating to this feature set relate to how well the tool supports each of the three main phases of an SR and the steps (or activities) within these phases. Here 12 questions will be asked. G4-F3-Q01 and G4-F3-Q02 concern features that support the planning phase of a SR. G4-F3-Q03 to G4-F3-Q09 relate to features supporting the conduct phase. G3-F3-Q10 and G3-F3-Q11 concern features that support the report phase.

G4-F3-Q01. How important is a feature that supports the development of a review protocol? (e.g. the tool provides support for collaboration using a template and control of versions to keep track of any changes to the protocol during its development.)

G4-F3-Q02. How important is a feature that supports protocol validation? (e.g. enabling evaluation checklists to be distributed to and completed by members of the review team.)

G4-F3-Q03. How important is a feature that provides support for the search process? (e.g. performing an automated search from within the tool which identifies duplicate papers and handles them accordingly.)

G4-F3-Q04. How important is a feature that provides support for study selection and validation? (e.g. the tool provides support for a multi-stage selection process, for multiple users to apply the inclusion/exclusion criteria independently and a facility to resolve disagreements.)

G4-F3-Q05. How important is a feature that provides support for quality assessment and validation? (e.g. the tool enables the use of a suitable quality assessment criteria, allows multiple users to perform the scoring independently and provides a facility to resolve conflicts.)

G4-F3-Q06. How important is a feature that provides support for data extraction? (e.g. the tool provides support for the extraction and storage of qualitative data using classification and mapping techniques and, in addition, the extraction of quantitative data, which manages the specific numerical data reported in a study, should also be supported.)

G4-F3-Q07. How important is a feature that provides support for data synthesis? (e.g. the tool provides automated analysis on extraction data such as table/chart generation.)

G4-F3-Q08. How important is a feature that provides text analysis?

G4-F3-Q09. How important is a feature that provides meta-analysis?

G4-F3-Q10. How important is a feature that supports the report phase of a SR? (e.g. the tool provides a template to assist the report write-up.)

G4-F3-Q11. How important is a feature that supports report validation? (e.g. automated evaluation checklists similar to the example given for protocol validation).

G4-F3-Q12. Are there any features you can think of that you might add to this feature set?

A5.4. Feature Set 4: (F4) process management G4-F4

Questions relating to this feature set relate to the management of an SR. Six questions will be asked.

G4-F4-Q01. How important is allowing multiple users to work on a single review?

G4-F4-Q02. How important are document management facilities? (e.g. in particular, managing large collections of papers, studies and the relationships between them.)

G4-F4-Q03. How important are security features? (e.g. log-in or a similar system.)

G4-F4-Q04. How important is the feature that provides support for role management? (e.g. state which users will perform certain activities, such as study selection, quality assessment, data extraction etc., and allocate papers accordingly.)

G4-F4-Q05. Is it important that the tool supports multiple projects? (i.e. the user can perform multiple SR projects using the tool.)

G4-F4-Q06. Are there any features you can think of that you might add to this feature set?

Appendix B. Interview Preparation Form

Each participant received the following information, sent on Keele University headed paper:

Study Title Tool Support for Systematic Reviews in Software Engineering

Aims of the Research The aim of this interview is to gather information about the availability, use, potential and effectiveness of automated tools which provide support for systematic reviews.

How long will the interview take? The interview should take no more than one hour to complete.

What will I be asked about? The interview will focus on discussing your thoughts and experience using tools to support the conduct of a systematic review. However, we are also interested in learning about the systematic review process particularly within your discipline. Questions will be asked in the following topics: The role of systematic reviews within your discipline. Known tools that are used to support the conduct of systematic reviews within your domain. Your personal experience undertaking systematic reviews (with/without the help of tools.)

How will information about me be used? The data collected will contribute towards the development of a refined framework for an overall tool to support SRs.

Who will have access to the information about me? The only people who will have access to the data collected are the members

of the research team conducting this study. This include Christopher Marshall (PhD Researcher), Prof Pearl Brereton (Lead Supervisor) and Prof Barbara Kitchenham (Second Supervisor). All data will be made anonymous during the analysis process for future reports and research projects. Notes taken during the interview process will be stored on a password protected computer. Audio recordings (providing you have agreed for the interview to be recorded) will be stored in a locked filing cabinet.

Who is funding the research? This research is partly supported by Keele University's Environmental, Physical Sciences and Applied Mathematics (EPSAM) Research Institute.

Appendix C. Coding participants comments about the lifecycle tool features

The mechanism used for coding the participants comments about specific features was to tabulate the comments each participant made about the feature. Then participant comments that addressed a general issue were highlighted, including comments:

- Identified benefits that the feature would deliver (Inc1).
- Identified possible problems or limitations associated with the feature (Inc2).

but excluding comments that:

- Restated or emphasized the participant's rating of the importance of the feature (Exc1).
- Discussed how the feature would work (Exc2).
- Restated some comment about the feature that had already been coded for that participant (Exc3).

The highlighted comments were read and the topics that addressed the same issue were identified and given a short description. The 22 features were coded one feature at a time. However, the use of codes was checked, so that if any similar comments occurred in subsequent features, the same terms were used. After the initial coding of features was completed, we reviewed single comments in each feature to investigate whether such comments occurred for different features.

The coding process was performed by Kitchenham using the comments tabulated by Marshall and then validated by Brereton.

For example, for the comment for the Search Process were as follows:

- P01
  - No comments.
- P02
  - That would be absolutely fantastic. (Comment ignored Exc1 – restated participants' rating of feature.)
- P03
  - That would *save a lot of time.* (Comment Inc1 coded as *Time Saving.*)
  - As long as the *process is done thoroughly and you're not missing anything.* (Comment coded Inc1 as *Viability* defined as 'will the feature work'?)
- P04
  - That would be brilliant. (Comment ignored Exc1.)
  - That would be *time saving.* (Comment coded as *Time Saving.*)
  - I'm not going to say anything is Mandatory I think, because I do them [SRs] without [the features]. (Comment ignored Exc1.)
- P05
  - *I can see there might be problems with that.* (Comment Inc2 coded as *Viability.*)
  - What might be good instead would be to help build this search strategy. (Comment Inc1 coded as *Help Search Strategy.*)
- P06
  - I mean it sounds highly desirable, but it sounds like quite a task. (Comment ignored Exc1.)
  - I think that as a reviewer, you'd probably want to see how they'd *actually confirmed that [that the feature worked].* (Comment Inc2 coded as *Viability.*)
  - I think if that was shown to be highly reliable it would be highly desirable. (Comment ignored Exc3 – restated previously coded comment.)
  - These search engines are updated regularly, These search engines are updated regularly, constantly update it [the feature]. (Comment ignored Exc3.)

  - It's a big ask. (Comment ignored Exc3.)
- P07
  - That's clearly mandatory in my book. That would be amazing. (Comment ignored Exc1.)
- P08
  - I think it could be useful to give an idea of the number of hits from each database. (Comment ignored Exc2 – discussing how the feature would work.)
  - It would help for a pilot search. (Comment ignored Exc2.)
  - I wouldn't want it to replace searching each individual database. (Comment ignored Exc2.)
  - I'm a bit against one search across all of the databases, because you are not actually searching the databases properly; you are not getting the best out of the databases. (Comment ignored Exc2.)
  - You would use *different strategies for different databases for good reasons.* (Comment Inc2 coded as *Viability.*)
- P09
  - *Well if it did it reliably.* (Comment Inc2 coded as *Viability.*)
  - The problem is you've got different controlled vocabularies in different databases. (Comment ignored Exc3.)
  - It would be highly difficult to automate all that. (Comment ignored Exc3.)
  - I think there are too many things in the way at the moment to be able to implement it. (Comment ignored Exc3.)
- P10
  - No comments.
- P11
  - No comments.
- P12
  - I would say highly desirable *but I don't trust you'd do it. I think there would be stuff missing.* (Comment Inc2 coded as *Viability.*)
- P13
  - Particularly about translating the search strategy. (Code ignored Exc1.)
  - I'd say that's highly desirable because *it's the thing that is time consuming.* (Comment Inc2 coded as *Time Saving.*)
  - The bit where our time is valuable is *most valuable is developing the search strategy in the first place. That sort of translating bit is very time consuming but it does not actually have to use that much expertise really.* (Comment Inc1 coded as *Help Search Strategy.*)

e-Informatica Software Engineering Journal, Volume 12, Issue 1, 2018, pages: 117–131, DOI 10.5277/e-Inf180105

Are We Working Well with Others? How the Multi Team Systems Impact Software Quality

Mathieu Lavallée*, Pierre N. Robillard*
*Département de génie informatique et génie logiciel, Polytechnique Montréal
mathieu.lavallee@polymtl.ca, pierre.robillard@polymtl.ca

Abstract

Background: There are many studies on software development teams, but few about the interactions between teams. Current findings suggest that these multi-team systems may have a significant impact on software development projects.
Aim: The objective of this exploratory study is to provide more evidence on multi-team systems in software engineering and identify challenges with a potential impact on software quality.
Method: A non-participatory approach was used to collect data on one development project within a large telecommunication organization. Verbal interactions between team members were analyzed using a coding scheme following the Grounded Theory approach.
Results: The results show that the interactions between teams are often technical in nature, outlining technical dependencies between departments, external providers, and even clients.
Conclusion: This article hypothesizes that managers of large software project should (1) identify external teams most likely to interfere with their development work, (2) appoint brokers to redirect external requests to the appropriate resource, and (3) ensure that there are opportunities to discuss technical issues at the multi-team level. Failure to do so could results in delays and the persistence of codebase-wide issues.

Keywords: multi team system, human interaction, quality management, team management, industrial study

1. Introduction

Five hundred years ago, John Donne wrote that "no man is an island". Individuals achieve great things by working together as a team. But many projects require more than an individual team to achieve success. "No team is an island" [1] would be a better description of modern project and organization management.

Teamwork has indeed long been identified as important to project success [2–4]. Teamwork in software development is no different, and software engineering research also highlighted the impacts that software development teams can have. As Watts S. Humphrey wrote, "Systems development is a team activity, and the effectiveness of the team largely determines the quality of the engineering" [5, p. 51]. Teams rarely work in isolation; teams are often interdependent of each other and must work together. Recent studies have shown the importance of these interactions between teams, whether on issues such as organization-wide knowledge sharing [6], coordination of multiple agile teams [7] or inter-team communication effectiveness [8].

This paper presents insights gained from the analysis of data collected in an exploratory study. These insights confirm the large amount of inter-team interactions, and identifies which teams were more closely connected to the development team. It also shows the role the developers play as middlemen between teams, for example between clients and testers. Finally, this study presents the importance of inter-team technical coordination, which is difficult if the organization

Table 1. Software engineering publications related to MTS in chronological order

Ref	Title (publication year)
[9]	Using open spaces to resolve cross team issue (2005)
[10]	Implementing Scrum in a distributed software development organization (2007)
[11]	Forming to performing: Transitioning large-scale project into Agile (2008)
[12]	Fully distributed Scrum: Replicating local productivity and quality with offshore teams (2009)
[13]	Moving back to Scrum and scaling to Scrum of Scrums in less than one year (2011)
[14]	Scaling Scrum in a large distributed project (2011)
[15]	Scrum practice mitigation of global software development coordination challenges: A distinctive advantage? (2012)
[16]	Coordination in co-located Agile software development projects (2012)
[17]	Practical Scrum-Scrum team: Way to produce successful and quality software (2013)
[18]	Coordination in large-scale Agile software development: A multiteam systems perspective (2014)
[6]	Fostering effective inter-team knowledge sharing in Agile software development (2015)
[19]	The effects of team backlog dependencies on Agile multiteam systems: A graph theoretical approach (2015)
[20]	A multiple case study on the inter-group interaction speed in large, embedded software companies employing Agile (2016)
[21]	The architect's role in community shepherding (2016)

only supports inter-team administrative coordination (i.e. resource planning and scheduling).

The next section presents the related work (Section 2), with a focus on the organizational psychology concept of multi-team systems and how it applies to software engineering. The methodology (Section 3) presents the context of the study and how the data was collected and analyzed. The results (Section 4) presents the data analysis, while the discussion (Section 5) presents our hypotheses and limitations to the conclusions of the study. The conclusion (Section 6) summarizes the hypotheses and presents future avenues of research. Note that this paper represents an extension of a previous shorter publication [22]. Some elements of the methodology were reused here, but the results and analyses are new.

2. Related work

The current software engineering literature uses different terms to define the interactions between multiple teams: inter-team, multi-team, cross-team, etc. However, these concepts are not always clearly defined, leaving the exact interpretation to the reader. The research field of organizational psychology has fortunately studied this topic extensively, regrouping them under the umbrella of multi-team systems, or MTS [23]. The MTS are defined as:

> Two or more teams that interface directly and interdependently in response to environmental contingencies toward the accomplishment of collective goals. MTS boundaries are defined by virtue of the fact that all teams within the system, while pursuing different proximal goals (e.g. writing a specific code module), share at least one common distal goal (e.g. creating a complete working software); and in so doing exhibit input, process, and outcome interdependence with at least one other team in the system [24].

Many studies have been published on single team dynamics in recent decades. Additionally, there is also a large body of knowledge on global or distributed software engineering, that is, multi-team systems spanning different sites across the globe. However, as far as we could find, there are few publications on the dynamics between co-localized teams. What should be done to make teams work together effectively within the same site at the organizational level?

> What is required for success in these kinds of MTSs is coordination both *within* and *between* teams [emphasis theirs]. That is, al-

> though interventions designed to create a system of strong, cohesive component teams may maximize performance at the team level, when ultimate system-level goals require synchronization between teams, more is needed. [...] MTS interventions must also address interdependencies between teams if performance across these kinds of complex systems is to be maximized [25].

Studies observing MTS in software engineering are still limited [18], with almost all studies found limited to Agile contexts and Scrum-of-Scrums meetings, as shown in Table 1.

Mike Cohn, an expert on the Scrum process, recommends a specific point in the agenda of "Scrum of Scrums" meeting, his version of MTS status meetings. Cohn recommends the addition of a question saying: "Are you about to put something in another team's way?" [26]. Cohn's recommendation outlines the importance of MTS and the impact one team can have on another. This recommendation was used in the field within "Scrum-of-Scrums" meetings, but with limited success [7]:

> Both case projects started using a model in which only one issue was discussed: impediments. However, this solution did not turn out well.[...] Both case projects still recognized the need for project-wide inter-team synchronization, but did not have any good solutions to the problem [7].

This shows that while the challenges of MTS projects are beginning to be better known, working solutions are still being tested [21].

2.1. Known challenges of MTS projects

This section presents a non-exhaustive list of challenges of MTS projects, based on what could be found in the literature. These three challenges were found to be most prevalent in the context of this study:

- Finding a compromise between team-level goals and MTS-level goals.
- Enabling effective communications and technical knowledge exchange at the MTS level,
- Planning the work at the MTS level.

One of the main MTS challenge is related to building a compromise between the objective of the local team goal and the overall goals of the MTS. In one software engineering case, the conflicting agendas of team members within different departments led to the failure of the project [13]. This challenge has a major impact on resource allocation. Organizational psychology researchers observed that "having to simultaneously work toward team-level goals along with MTS-level goals creates a demanding work environment" [25]. In software engineering, Santos et al., reached a similar conclusion. They studied knowledge sharing between teams in an Agile context [6]. They noted that the introduction of new MTS support practices requires more resources, which must be provided by the organization, otherwise the practice, and potentially the project, could fail.

Another challenge is the relative difficulty to ensure efficient communications at the MTS level, compared to communications within the team. A survey conducted by Kiani et al. noted that due to "lack of communication, almost fourth of respondents complained that work items they depended on have changed without any notification" [27]. Some basic Agile principles are also affected in MTS contexts. For example, face-to-face communications are easy at the team level, but are difficult to apply at the MTS level. It requires the organization to mix people from one team to another, which is not always possible [28]. "Boundary spanning", ensuring communications between the frontiers of the teams, is an important challenge within MTS [16, 20].

In the same vein, dissemination of technical information specific to a field of knowledge is also difficult. Local teams accumulate a significant amount of knowledge about the specific area in which they work. How can this knowledge be effectively communicated to the other teams in the MTS? If the project is particularly complex, it may also be difficult to get an overall view of the project [14]. Each team knows its own problems, which can be difficult to translate in a form understandable by other teams that might not have the same knowledge of the field.

A third challenge is related to how MTS coordination should be planned. Lanaj et al. found the following.

> Decentralized planning has positive effects on multiteam system performance, attributable to enhanced proactivity and aspiration levels. However, [...] the positive effects associated with decentralized planning are offset by the even stronger negative effects attributable to excessive risk seeking and coordination failures [29].

The study of MTS coordination has been identified by one study as "underdeveloped" [18]. However, MTS is a concept defined within the domain of organizational psychology. Research in software development already has a large body of knowledge pertaining to inter-team interactions within the domain of global and distributed software development [30]. While a global or distributed development team is a form of MTS, some MTS can be collocated in the same building. The team observed interacted with other teams which were almost all collocated within the same building. The context of this study is therefore different from the study of global and distributed software development, where the issues of geographical distance and temporal distance play a large role.

3. Methods

3.1. Industrial context

The study was performed on a large telecommunications organization with over forty years of experience in the industry. Throughout the years, the organization has developed a large codebase, which must be constantly updated. This study follows one such update project. The outcomes of this study are based on ten months of observation of a software development team involved in a two-year project for an internal client. The project involved a complete redesign of an existing software package used in the organization's internal business processes.

The technical challenge of this update project is that it requires the modification of COBOL legacy software, Web interfaces, mobile device integration and multiple databases. Its purpose is to manage work orders. To do this, it needs to extract data from multiple sources within the enterprise (employee list, equipment list, etc.) and send it to multiple databases (payroll, quality control, etc.).

The project was a second attempt to overhaul this complex package. A first attempt had been made between 2010 and 2012 but was abandoned after the fully integrated software did not work. Because this project was a second attempt, many specifications and design documents could be reused. Accordingly, the development followed a traditional waterfall process, as few problems were expected the second time around. This second attempt began in 2013 and was successfully deployed during October and November 2014.

The organization has no formal MTS coordination practices in place. Coordination at the MTS level is therefore mostly tacit. This means that when a team needs information from another team, a member of the first team has to directly contact another member of the second team. This causes some issues at the MTS level, because most developers in the team observed were new to the company [31] when the project started, and in some cases did not know who to contact in the other teams. Despite its tacit nature, an MTS exists. The need for coordination between the projects means that interactions between teams are required to perform the work.

This study observes a development team of nine members: one manager, four senior developers, two junior developers and two contract developers. The team was formed specifically for this project, of which seven are new to the organization (i.e. less than five years).

Note that the nature of this MTS is different from an MTS where several teams are working on the same project (e.g. a Scrum-of-Scrums development project). In the MTS observed, all teams had different projects, with their own goals and objectives. The development project studied was the responsibility of a single team, the team observed. However, to perform that project, that team could not do it alone, and had to seek help from other teams.

The objective of this study is to understand how a development team interacts with other external team to do its work. Therefore, the focus is on the development team. Who does the development team needs to talk to and why?

3.2. Study approach

The objective of the study was to identify the cause behind the introduction of quality problems during software development. Given the sensitive nature of problem identification within a large organization, it was decided to opt for a neutral approach. Data collection was to be performed using a non-participatory approach, to avoid organizational influence.

Data collection was limited to weekly status meetings because that is the avenue used by the organization to discuss and resolve MTS issues. Although there were certainly discussions between teams outside these weekly status meetings, the most important issues were discussed at these meetings.

A qualitative approach was chosen to better understand an area where many variables are not fully identified. The approach of this study uses the same rationale as Looney and Nissen:

> The present research is exploratory in nature, is not guided by extensive theory, and is approaching a "how" research question. Hence qualitative field research reflects an appropriate method [32].

3.3. Data collection methodology

This study is based on non-participant observation of the software development team's weekly status meetings. These meetings consisted of mandatory all-hands discussions for the eight developers assigned mostly full-time to the project, along with the project manager. These meetings included, as needed, developers from related external modules, testers, database administrators, security experts, quality control specialists, etc. The meetings involved up to 15 participants, and up to five additional participants through conference calls.

The team discussed the progress made during the previous week, the work planned for the coming week and obstacles to progress. The problems raised concerned resources and technical issues. Few decisions were taken at these meetings, the purpose being to share the content of the previous week's discussions between the different teams.

A round-table format was used, where each participant was asked to report on their activities. The discussions were open and everyone was encouraged to contribute. When a particular issue required too much time, participants were asked to set another meeting to discuss it. Meetings lasted about an hour.

The data presented in this study was collected over seven months during the last phase of the two-year project. It is based on 21 meetings held between January and July 2014. The same observer attended all the meetings and took note of who was involved in each interaction, the topic being discussed, and the outcome. A typical interaction would last between 5 and 30 seconds. The notes were then produced as quasi-verbatim transcripts.

3.4. Coding methodology

Due to the large amount of data collected, it is necessary to summarize the data obtained in order to find patterns. This summarization was performed using a coding methodology based on the grounded theory approach [33].

Coding was performed after the observations were completed, based on the meeting notes taken from February 27th, 2014 to July 31st, 2014. Since it can take time for the people observed to be used to the presence of the researcher [34], and for the researchers themselves to fully understand the domain knowledge of the project [35], the data from the first two meetings were not kept for this study.

Meetings taking place after July 31st were also removed from this analysis. These last meetings were mostly related to deployment activities and featured very little development interactions. While the analysis of the deployment activities

would be interesting, it was decided to keep the development discipline and deployment discipline separate, as the MTS requirements of both disciplines are quite different.

Coding schemes were developed following the Grounded theory approach [33]. In summary, coding was performed using the following steps:

1. Open coding of all entries, going over the data as long as new codes can be added.
2. When no new codes can be added, similar codes are grouped together.
3. Code groups are formalized into schemes.
4. Return to point (1) until no new codes are added and no new schemes can be formed.

After multiple coding iterations, three coding scheme emerged. The first scheme pertains to whether the interaction observation is related to a technical or administrative topic:

Technical: interactions related to technical issues (requirements, bugs, data, etc.),

Administrative: interactions related to administrative issues (deadlines, resources, etc.).

The second scheme pertains to one of the four types of interaction identified:

Team demands (inputs): These interactions are requests made by team members to someone outside the development team.

Team commitments (outputs): These interactions are requests made by someone outside the development team to the team or a team member.

Team coordination (in-out): These interactions are related to meetings which had or will take place between two or more teams on a given issue.

Team liaison (brokering): These interactions are information request to the development team by someone outside the team. The development team cannot answer themselves and therefore act as knowledge brokers with another team.

The third scheme pertains to the type of team interacted with:

Client teams: These teams are responsible for providing requirements and details on what they need the software to do, along with validation of the final result.

3rd party teams: These teams represent the 3rd party library support teams, which performs corrections on the software based on the service-level agreement (SLA) their 3rd party holds with the organization. Two internal module support teams are also included here, as the interaction with these teams followed a protocol similar to the interaction with support teams outside the organization.

Quality teams: These teams are responsible for quality assurance and quality control within the organization.

Ancillary teams within the organization: The organization has many departments, each with their own expertise and technical competencies. For example, one ancillary team was in charge of the creation and configuration of the development and test environments.

In-house development teams: These teams represent other development teams working in parallel projects on the same codebase.

4. Analysis

Data collection returned a total of 464 topics discussed within the 21 weekly status meetings analysed. From these 464 topics, 294 were related to external teams. Therefore, about 60% of all topics discussed were related to requests to external teams, commitments to fulfil for stakeholders, and other interactions that involved external team members.

Figure 1 presents the number of interactions between the observed development team and all external teams. The teams are split based on the five team types presented in the previous section. The closer a team is to the dark centre of Figure 1, the more interactions they had with the observed team, and the closer they were to them. Note that since it was an internal development project for an organization which does not sell software, the actual clients of the package upgraded was the Operations team. The Operations team is in charge of creating and dispatching work orders. Field workers receive the work orders and must on occasion interact with the software. A total

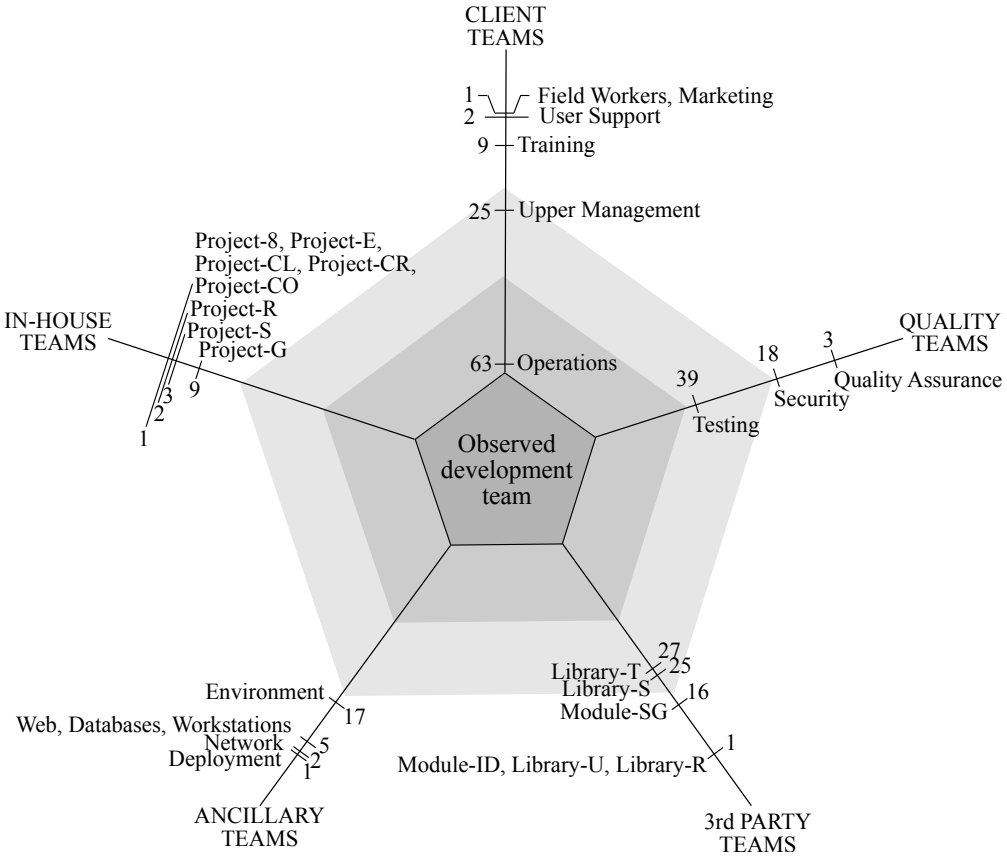


Figure 1. Proximity of each external team with the observed development team. The number of interactions are posted on the axes

of 29 different external teams were contacted during the course of the study.

Table 2 presents the results of the number of interactions with external team members according to their activities, which outlines the amount of interactions and the rationale for interaction (to answer team needs, to fulfil team obligations, etc.). While table 2 present the number of liaison interactions, more details are presented in Figure 2. Table 2 shows that there are numerous administrative as well as technical interactions with all the team categories. Note that eight interactions could not be assigned to a specific team, bringing the total in Table 2 to 294 interactions.

Figure 2 shows the occurrences of liaison interactions between two teams in which the observed development team was involved. It shows that the observed team is pivotal between the client and the quality group. These interactions include requirement clarifications, but also demands by testers to ensure that the initial data in the system are validated by the clients before testing can start. More details

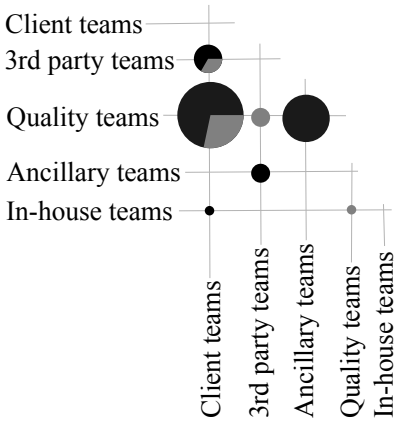


Figure 2. Liaison interactions (knowledge brokering) between external team categories. Bubble size represents the amount of liaison interactions (from one to seven). Black colour represents technical interactions, while grey colour represents administrative interactions

Table 2. Number of interactions with external teams per team category

Team Category	Team Demands		Team Commitments		Team Coordination		Team Liaison		Total
	Tech	Admin	Tech	Admin	Tech	Admin	Tech	Admin	
Client teams	22	21	19	11	8	7	8	5	101
3rd party teams	35	7	8	3	6	4	4	4	71
Quality teams	3	6	19	7	6	4	10	5	60
Ancillary teams	14	8	3	1	2	0	7	0	35
In-house teams	4	1	8	2	1	1	1	1	19

about the importance of the client/quality interactions can be found in the next section.

4.1. Interaction purpose examples

Tables 3, 4, 5 and 6 present a glimpse of the reasons through actual quotes from the development team. Each table covers one of the four types of interactions. The objective is to give an idea how a topic was associated with the appropriate interaction type and the appropriate external team.

4.2. Failure of the first iteration of the project

As stated earlier, the project observed had already been done once, but failed. A private communication with a manager who witnessed the failure of the first iteration but did not participate in the second one provides some details on the failure. According to the manager, the following factors may have caused the failure of the first iteration:

- Personality conflicts between the development team, the client teams, and the other 3rd party teams. This can be related to "Organizational Skirmish" identified by Tamburri et al. [36].
- Contractual issues between the organization and 3rd party developers. Contract negotiations dragged so long that the contracts were signed moments before the code was scheduled for production.
- Pressure from the project manager to filter interactions with the development team. This manager required that all requests had to be submitted directly to her, resulting in missed or misinterpreted messages. This can be related to the "Radio-Silence" identified by Tamburri et al. [36].
- Documentation mostly incomprehensible by anyone outside the development team. Only the client teams' documentation could be reused as is.

While this statement is only supported by one witness, it still provides some insight as to why the project initially failed.

5. Discussion

This section discusses the results and poses three hypotheses to resolve the identified issues, along with their potential impact on quality.

5.1. First hypothesis: Identification of the critical teams and client implication

This study shows that although interactions with external teams are important, some teams are more important than others. The frequency analysis shows that the interactions of the team loosely follow a Pareto distribution. Approximately 78% of external interactions (229 of 294 interactions) are made from about 28% of all teams contacted (8 of 29 teams). Based on the data in Figure 1, the distribution of these eight teams (categories of the corresponding team in brackets) are:

1. Operations [client team]: 63 interactions.
2. Testing [quality team]: 39 interactions.
3. Library-T [3rd party team]: 27 interactions.
4. Library-S [3rd party team]: 25 interactions.
5. Upper Management [client team]: 25 interactions.
6. Security [quality team]: 18 interactions.

Table 3. Example quotes related to team demands

Demands to	Quote
Client	*"There is a problem with [the client]. We need the configuration data and we have no answer from [the client]. I did some work on this, but I cannot finish by myself."* The team had to ask the client again for the configuration data.
Ancillary	*"Everything has been settled, except for the database configuration. We do not have the access rights [to the environment] to prepare this. [...] This configuration should be done by default! It's like buying a car and not having a key!"* The team had to ask the environment setup team for the rights to change the database configuration.
In-House	*"We just receive an analysis from Project-G, which is about 60 pages. The analysis is very badly written and is essentially incomprehensible."* The team had to ask the Project-G team a clearer document in order to fulfil the analysis.

Table 4. Example quotes related to team commitments

Commitments to	Quote
Client	Upper Management has approved a new project with a high priority and a very aggressive calendar. It is likely that some developers from the development team will be assigned to this new project. The observed development team must finish their current project as soon as possible, as delays will be unacceptable for upper management.
Quality	*"What do we do if we find bugs?"* Quality teams need development support during the developers' holiday, in August. The development team cannot go on holiday all at once: someone must stay in place to correct the bugs found by quality teams.
In-House	The development team must replace a function so it can support true/false/maybe values. This is in order to support Project-R, developed by another team, which will be deployed shortly after their current project ends.

Table 5. Example quotes related to team coordination

Coordination with	Quote
Client	The development team needs the business processes from the client so they can code the appropriate functionalities. But the client expects that the development team will explain how the software will work, and therefore adjust their business process in consequence. There is confusion as to whom is responsible for providing the business processes.
3rd Party	The development team must discuss with Library-T support to determine which changes will be covered under the current contract and which changes will be charged extra to the project.
Quality	The development team pressures the testing team to start acceptance testing even though integrated testing is not finished. The testing team disagrees: the two teams will need to meet afterward in order to decide what to do. *"How can I start acceptance testing if integrated testing only reach 50% success?"*

7. Environment [ancillary team]: 17 interactions.
8. Module-SG [3rd party team]: 16 interactions.

While the other 21 teams have less than ten interactions each.

Therefore, project managers should try to identify the teams most likely to have an impact on the project beforehand, and ensure that communication channels with these teams are clear. In the case observed, this issue was somewhat alleviated by making the testing team sits in the same room as the developers towards the end of the project. They could not do the same with their 3rd party developers, which resulted in some serious issues. For example, communication problems with 3rd party support teams, coupled

Table 6. Example quotes related to team liaison

Liaison between	Quote
From client to quality	The clients need to provide a description of their workflow for the testing team. The testing team are planning acceptance testing and want to design tests which reflect what the client does in its day-to-day work.
From quality to client	A client was assigned to the testing team in order to assist them in their work. However, the client assigned does not answer the telephone or email. The quality team needs to talk to him.
From quality to ancillary	The security team need access to the test environment in order to perform their tests. The Network team needs to open a port for the security team.
From in-house to quality	Testers need to know if they need to perform testing for the integration of Project-G within the current project. So far, the in-house team developing Project-G has not answered.

with poor service-level agreements (SLA), required multiple reworks of some simple change requests, each taking one month to perform [22].

The Pareto analysis shows that the clients, the Operations team, is by far the external team most contacted. However, the development project followed a waterfall approach, with fixed requirements. Why so many interactions are needed with the clients if the requirements are fixed since the beginning? Many details and subtleties became evident as the developers progressed into the project. Some requirements have emerged or have changed very late during the project. Some of these changes were client requests, but others were tasks that the client needed to do.

For example, since this project is related to the update of an old package, some of the new databases must be updated with the data already in the old package. However, a lot of the data in the old package are obsolete: dropdown menu items are no longer used; database columns are no longer filled, etc. The developers cannot know these subtleties, and rely on clients to tell them which data to port to the new package, and which data to remove. In this case, the clients did not have the resources to do this task for the developers, which leads to multiple delays.

This shows that all projects, whether Agile or disciplined, require continuous interactions with stakeholders. But while Agile principles emphasize flexibility to clients' needs ("our highest priority is to satisfy the customer" [37]), this study shows that the clients must also be flexible to developers. Clients have obligations to fulfill. The domain knowledge of the clients was very important in this project. Some delays can be attributed to the unavailability of the client or to late responses to critical requests. The clients were required to provide many details about what the old package did, and why the old package worked that way, and on what the client wants for the new package. The clients' technical expertise was limited, but they knew very well their workflow and how they want the future application to merge with this workflow.

For project managers, we propose that any client/provider agreement ensures that the client is willing to actively help developers. For example, the simple task of seeding a database with its initial dataset is difficult to plan ahead: it can be done once the database structure is completed, using data that is usually provided by the client. In this study, delays in obtaining clients responses have led to delays in database configuration, which caused the tests to start late, and ultimately to be shorter than planned.

Previous Agile studies have used client delegates to ensure coordination between the real client and the development teams [11, 14]. Delegation of client duties can prevent constant interruption of the workflow of developers. One study assigned each team with "a support person".

> Supporting the customer using all the solutions that the team had to provide was a critical task. This required a vast amount of knowledge of all moving pieces. Before we had the support person, the customers interrupted subject matter experts [i.e. developers] directly. The subject matter experts typically

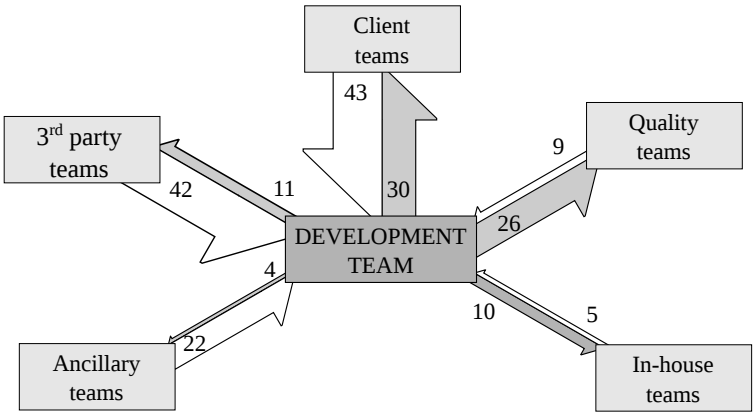


Figure 3. Chain of commitments between teams. White arrows indicate answers to development team demands. Grey arrows indicate team commitments that the development team must fulfil

dealt with too many support requests and ended up context switching in and out of the tasks at hand [11].

The impact on quality in the case observed is mainly transcribed in terms of delays. The failure to provide answers in a due manner to the questions of the development team led to multiple delays. In this case, these delays cause the testing phase to be greatly reduced. In addition, some code written by 3rd parties could not be reviewed in time for delivery and was included in the codebase as-is. By the accounts of the developers themselves, a lengthy support process will be necessary post-delivery to ensure that all the issues are sufficiently smoothed out.

5.2. Second hypothesis: Developers as knowledge brokers within the MTS

Figure 3 illustrates the two-way interactions between external teams and the development team, based on the team demands and team commitments found in Table 2. For example, in the interactions between the development team and the client teams, the development team had 30 commitments (grey arrow) toward the client teams, while the client teams answered 43 demands (white arrow) from the development team.

The left hand side of Figure 3 shows the team categories with a majority of demands from the development team (large white arrows), while the right hand side shows the team categories with a majority of commitments from the development team (large grey arrows). The client teams, being fairly balanced in demands and commitments, remains in the middle. What should be seen from Figure 3 is that demands flows from the left to the right. Ancillary teams fulfil developers' requests, so that developers can fulfil quality teams' requests.

Here is an example taken from the interactions observed. The quality teams needed many test environments in order to perform their work (acceptance environment, load testing environment, etc.). The development team was therefore committed into building these environments and ensuring that they were coherent with the latest available versions of the package and that they were stable enough to support test activities. While they could do some of the work themselves, they needed the support of the environment setup team, an ancillary team. However, the environment setup team did not fulfil its commitment appropriately, causing a number of issues to the development team. These environmental issues cause the development team to fail in some of their commitments toward the quality teams, causing delays and ultimately, the cancellation of some of the test activities.

Some of these relationships might seem self-evident, but others might not be as well-known. As presented in our previous paper [22], managers should be wary of other projects imposing changes to the current project. Project managers should also ensure that all relevant teams (3rd party teams, ancillary teams) are ready to help the

development team. In the case presented above, many issues stemmed from poor communications between the development team and the environment team.

The role of the development team in this case is that of a broker. Developers need to redirect the requests they receive to the appropriate team. To take an analogy from the TCP/IP protocol, the development team is the default gateway for the external teams. External teams needing something related to the project will ask the developers first, which will then redirect the team to the appropriate resource when necessary.

This is especially true of the relationship between clients and testers. Clients and testers do not know how the application was built, who was contacted to code the software, what are the dependencies. They are mostly conscious on what they see on their end. When something goes wrong, their only contact is the development team. Clients and testers need some answers but do not know who to ask; developers know and must assist them.

For project managers, this study shows that testers cannot work efficiently if they are kept completely isolated from the development team. Testers need to ask many questions in order to perform their work, and these questions must be efficiently relayed to the appropriate external team. In the case studied, toward the end of the project, management had the testing team sits directly with the development team. Their goal was to diminish bug resolution times, but it also helped the testers in the setup of the different testing phases and testing environment (integration, acceptance, load, and deployment). The same can be applied to clients. While it might not make sense to put the client in contact with every relevant external team, clients' questions can be distributed by the developers to the relevant external teams.

The need for knowledge brokers have been identified in the literature [11, 25, 36]. It is sometimes identified as a "coordinator role" [16].

> Brokers are those individuals who link disconnected subgroups. [Another study] found that system-level coordination is achieved more efficiently when certain key individuals connect different subgroups as opposed to when all individuals are directly connected to one another. Complex MTSs may be more efficiently coordinated if certain individuals act as ambassadors by connecting their team to others within the system [25].

The question of whom to assign to the role of knowledge broker varies from study to study however. It should be someone who has a widespread knowledge of the system [10, 11]. It is, however, unnecessary to have a broker between each team. As presented in the previous section, teams with a potential critical impact on the development team's work should be identified. Knowledge brokers can therefore be assigned only for those critical teams [38]. Minor teams and modules could be more isolated from the development team under study.

The impact on quality rests on the fact that the development team does not work in isolation. There are many other teams working indirectly on the project which require adequate support to perform their work. Here are a few examples:

- Testers need to obtain real data from the clients in order to perform tests that can be relatable to what goes on in reality.
- Testers need working testing environment with an up-to-date code in order to perform adequate tests.
- Third party support teams need to know the type of tests to be performed in order to ensure that their infrastructure will support these tests (e.g. stress testing or security testing cloud storage services).

Failure to relay the needs of one external team to another can lead to the cancellation of important activities.

5.3. Third hypothesis: Managing technical and administrative interactions

Before this study, the organization managers and the development team were convinced that their meetings were mostly administrative; discussing deadlines, budget and resources. Observations proved that most of these discussions were actually technical and involved bugs, issues, design,

solutions, etc. It is therefore not surprising that most of the interactions with external teams are also technical in nature.

But this information should not be exchanged only from one manager to another. This study's suggestion to project managers is to make sure that developers in different teams are able to talk to each other. Managers have a tendency to protect their developers from outside interference, and it is good to keep an eye on that, as this was an issue with Upper Management in this case [22]. But developers also need to be able to obtain technical information from other teams, and to plan technical solutions and strategies together.

The literature recommends a layered structure where the lower levels are able to share technical details, while the higher levels are able to share the administrative big picture [10, 11].

> Cross team knowledge sharing is difficult. [...] After 1.5 year into practicing Agile, we found the best way to mitigate, is to have weekly Scrum of Scrums (S2) meetings and daily tech leads stand-up meeting. For the stakeholders, Scrum of Scrums of Scrums (S3) was very helpful to get things prioritized [11].

The impact on quality is that technical issues facing the whole codebase are not discussed anywhere. Individual teams might be aware of the issues, but without a platform to discuss and voice their concern, these issues remain latent and unaddressed. Organizations have administrative strategies, where managers discuss future plans and projects, but how many of them have technical strategies, where engineers can discuss future maintenance challenges and issues?

5.4. Threats to validity

A threat to the validity with the use of a single study is the generalizability of its conclusions. The objective of this study was however not to build a theory applicable to all software development projects, but to identify new potentially interesting practices and issues from the industry. While this study is limited to a single case, it nonetheless presents new qualitative and quantitative data showing the role of clients during development, the role of developers as knowledge brokers, and the importance of technical coordination at the MTS level.

Proper case study practices recommend triangulating the data, that is to obtain data from different approaches in order to confirm the conclusions [39, p. 97]. For instance, conclusions made through observations can be confirmed with interviews and artefact analyses. In this case, it was not possible to access any other data source, limiting the work to an exploratory study instead of a fully fledged case study. That is why the recommendations are presented as hypotheses to be tested, instead of solutions.

6. Conclusions and further works

This exploratory study shows the impact interactions within the multi team system can have on project success. Due to the single study nature of this research, future research should look into whether the three hypotheses presented herein are relevant in other cases.

1. Identify the external teams most likely to have an impact on the development project, based on a Pareto analysis and ensure proper communication channels with the most important ones. Otherwise, slow communications will cause delays during development, which might result in rush development work and shorter testing time.
2. Ensure that knowledge brokers exist within the development team to redirect requests from one external team to the proper other external team. Otherwise, some activities with an indirect impact on the development project (e.g. testing) might be in jeopardy.
3. Ensure that discussion platforms at the multi-team level are not limited to administrative issues. Technical solutions and strategies must be discussed between teams. Otherwise quality issues affecting the whole codebase could remain unaddressed.

Project managers should be aware of the impact of multi team systems on their projects.

From a disciplined, plan-driven approach, to an Agile, people-driven approach, there is a need for an integrated, organization-driven approach, where the team is integrated within its organization. Teamwork experts have recommended breaking the isolation between individuals in order to ensure that the whole team works together. We should now see if these recommendations hold at the organization level, in order to ensure that the whole organization works together. There might be no "I" in "team". But how much place for "us" and "them" are we willing to work with within the organization?

7. Acknowlegments

This research would not have been possible without the agreement of the company in which it was conducted, which prefers to stay anonymous, and without the generous participation and patience of the software development team members from whom the data were collected. To all these people, we extend our grateful thanks.

This work was supported by the Natural Sciences and Engineering Research Council of Canada, under grant number A-0141.

References

[1] J. Porck, *No Team is an Island: An Integrative View of Strategic Consensus between Groups*, Ph.D. dissertation, Erasmus University Rotterdam, 2013.

[2] F.Q. da Silva, A.C.C. França, M. Suassuna, L.M. de Sousa Mariz, I. Rossiley, R.C. de Miranda, T.B. Gouveia, C.V. Monteiro, E. Lucena, E.S. Cardozo, and E. Espindola, "Team building criteria in software projects: A mix-method replicated study," *Information and Software Technology*, Vol. 55, No. 7, 2013, pp. 1316–1340.

[3] R.A. Guzzo and M.W. Dickson, "Teams in organizations: Recent research on performance and effectiveness," *Annual Review of Psychology*, Vol. 47, 1996, pp. 307–338.

[4] R.A. Guzzo and E. Salas, *Team Effectiveness and Decision Making in Organizations*. Wiley, 1995.

[5] W.S. Humphrey, "The Team Software Process (TSP)," Software Engineering Institute, Pittsburgh, PA, USA, Tech. Rep. ESC-TR-2000-023, 2000. [Online]. https://www.sei.cmu.edu/reports/00tr023.pdf

[6] V. Santos, A. Goldman, and C.R.B. de Souza, "Fostering effective inter-team knowledge sharing in Agile software development," *Empirical Software Engineering*, Vol. 20, No. 4, 2015, pp. 1006–1051.

[7] M. Paasivaara, C. Lassenius, and V.T. Heikkilä, "Inter-team coordination in large-scale globally distributed Scrum: Do Scrum-of-Scrums really work?" in *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '12. New York, NY, USA: ACM, 2012, pp. 235–238.

[8] A. Martini, L. Pareto, and J. Bosch, *Improving Businesses Success by Managing Interactions among Agile Teams in Large Organizations*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 60–72.

[9] C.M. Tartaglia and P. Ramnath, "Using open spaces to resolve cross team issue [software development]," in *Agile Development Conference (ADC'05)*, 2005, pp. 173–179.

[10] H. Smits and G. Pshigoda, "Implementing scrum in a distributed software development organization," in *Agile 2007*, 2007, pp. 371–375.

[11] E.C. Lee, "Forming to performing: Transitioning large-scale project into Agile," in *Agile 2008 Conference*, 2008, pp. 106–111.

[12] J. Sutherland, G. Schoonheim, and M. Rijk, "Fully distributed Scrum: Replicating local productivity and quality with offshore teams," in *2009 42nd Hawaii International Conference on System Sciences*, 2009, pp. 1–8.

[13] R.P. Maranzato, M. Neubert, and P. Herculano, "Moving back to Scrum and scaling to Scrum of Scrums in less than one year," in *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion*. New York, NY, USA: ACM, 2011, pp. 125–130.

[14] M. Paasivaara and C. Lassenius, "Scaling Scrum in a large distributed project," in *2011 International Symposium on Empirical Software Engineering and Measurement*, 2011, pp. 363–367.

[15] P.L. Bannerman, E. Hossain, and R. Jeffery, "Scrum practice mitigation of global software development coordination challenges: A distinctive advantage?" in *2012 45th Hawaii International Conference on System Sciences*, 2012, pp. 5309–5318.

[16] D.E. Strode, S.L. Huff, B. Hope, and S. Link, "Coordination in co-located Agile software de-

velopment projects," *Journal of Systems and Software*, Vol. 85, No. 6, 2012, pp. 1222–1238, special Issue: Agile Development.

[17] A. Mundra, S. Misra, and C.A. Dhawale, "Practical Scrum-Scrum team: Way to produce successful and quality software," in *2013 13th International Conference on Computational Science and Its Applications*, 2013, pp. 119–123.

[18] A. Scheerer, T. Hildenbrand, and T. Kude, "Coordination in large-scale Agile software development: A multiteam systems perspective," in *2014 47th Hawaii International Conference on System Sciences*, 2014, pp. 4780–4788.

[19] A. Scheerer, S. Bick, T. Hildenbrand, and A. Heinzl, "The effects of team backlog dependencies on Agile multiteam systems: A graph theoretical approach," in *2015 48th Hawaii International Conference on System Sciences*, 2015, pp. 5124–5132.

[20] A. Martini, L. Pareto, and J. Bosch, "A multiple case study on the inter-group interaction speed in large, embedded software companies employing Agile," *Journal of Software: Evolution and Process*, Vol. 28, No. 1, 2016, pp. 4–26, jSME-14-0083.R3.

[21] D.A. Tamburri, R. Kazman, and H. Fahimi, "The architect's role in community shepherding," *IEEE Software*, Vol. 33, No. 6, 2016, pp. 70–79.

[22] M. Lavallée and P.N. Robillard, "Why good developers write bad code: An observational case study of the impacts of organizational factors on software quality," in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 1, 2015, pp. 677–687.

[23] M.A. Marks, L.A. DeChurch, J.E. Mathieu, F.J. Panzer, and A. Alonso, "Teamwork in multiteam systems," *Journal of Applied Psychology*, Vol. 90, No. 5, 2005, pp. 964–971.

[24] J.E. Mathieu, M.A. Marks, and S.J. Zaccaro, *Multi-team systems*. London: Sage, 2001, pp. 289–313.

[25] R. Asencio, D.R. Carter, L.A. DeChurch, S.J. Zaccaro, and S.M. Fiore, "Charting a course for collaboration: A multiteam perspective," *Translational Behavioral Medicine*, Vol. 2, No. 4, 2012, pp. 487–494.

[26] M. Cohn, *Advice on Conducting the Scrum of Scrums Meeting*, 2007. [Online]. https://www.scrumalliance.org/community/articles/2007/may/advice-on-conducting-the-scrum-of-scrums-meeting Retrieved 2015-08-21.

[27] Z.U.R. Kiani, D. Smite, and A. Riaz, "Measuring awareness in cross-team collaborations – Distance matters," in *2013 IEEE 8th International Conference on Global Software Engineering*, 2013, pp. 71–79.

[28] T. Chau and F. Maurer, *Knowledge Sharing in Agile Software Teams*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 173–183.

[29] K. Lanaj, J.R. Hollenbeck, D.R. Ilgen, C.M. Barnes, and S.J. Harmon, "The double-edged sword of decentralized planning in multiteam systems," *Academy of Management Journal*, Vol. 56, No. 3, 2013, pp. 735–757.

[30] J.M. Verner, O.P. Brereton, B.A. Kitchenham, M. Turner, and M. Niazi, "Systematic literature reviews in global software development: A tertiary study," in *16th International Conference on Evaluation Assessment in Software Engineering (EASE 2012)*, 2012, pp. 2–11.

[31] M. Lavallée and P.N. Robillard, in *2015 IEEE/ACM 3rd International Workshop on Conducting Empirical Studies in Industry*, 2015, pp. 12–18.

[32] J.P. Looney and M.E. Nissen, "Organizational metacognition: The importance of knowing the knowledge network," in *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, 2007, p. 190c.

[33] A.L. Strauss, *Qualitative Analysis for Social Scientists*. Cambridge University Press, 2003.

[34] H.A. Landsberger, *Hawthorne Revisited*. Cornell University, 1958.

[35] T.C. Lethbridge, S.E. Sim, and J. Singer, "Studying software engineers: Data collection techniques for software field studies," *Empirical Software Engineering*, Vol. 10, No. 3, 2005, pp. 311–341.

[36] D.A. Tamburri, P. Kruchten, P. Lago, and H. van Vliet, "Social debt in software engineering: Insights from industry," *Journal of Internet Services and Applications*, Vol. 6, No. 1, 2015, p. 10. [Online]. https://jisajournal.springeropen.com/articles/10.1186/s13174-015-0024-6

[37] P. Runeson, A. Stefik, and A. Andrews, "Variation factors in the design and analysis of replicated controlled experiments," *Empirical Software Engineering*, Vol. 19, No. 6, 2014, pp. 1781–1808.

[38] R.B. Davison, J.R. Hollenbeck, C.M. Barnes, D.J. Sleesman, and D.R. Ilgen, "Coordinated action in multiteam systems," *Journal of Applied Psychology*, Vol. 97, No. 4, 2012, pp. 808–824.

[39] R.K. Yin, *Case Study Research: Design and Methods*, ser. Applied Social Research Methods, L. Bichman and D.J. Rog, Eds. Thousand Oaks, CA, USA: Sage, 2002, Vol. 5.

e-Informatica Software Engineering Journal, Volume 12, Issue 1, 2018, pages: 133–165, DOI 10.5277/e-Inf180106

A Systematic Mapping Study on Software Measurement Programs in SMEs

Touseef Tahir*, Ghulam Rasool*, Muhammad Noman*
*COMSATS Institute of IT, Department of Computer Science, Lahore, Pakistan.
touseeftahir@ciitlahore.edu.pk, grasool@ciitlahore.edu.pk, nomanyasir29@gmail.com

Abstract

Context: Software measurement programs are essential to understand, evaluate, improve and predict the software processes, products and resources. However, the successful implementation of software measurement programs (MPs) in small and medium enterprises (SMEs) is challenging.
Objective: To perform a detailed analysis of studies on MPs for highlighting the existing measurement models, tools, metrics selection methods and challenges for implementing MPs in SMEs.
Methods: A Systematic Mapping Study (SMS) is conducted.
Results: In total, 35 primary studies are comprehensively analysed. We identified 29 software measurement models and 4 tools specifically designed for MPs in SMEs. The majority of the measurement models (51%) are built upon software process improvement approaches. With respect to the measurement purposes of models, the distribution of MPs was identified as: characterization (63%), evaluation (83%), improvement (93%) and prediction (16%). The majority of primary studies discussed the use of measurement experts and experience (60%) followed by the use of measurement standards (40%) and the use of automated tools (22%) for metrics selection in MPs. It was found that the SMEs and large organizations face different challenges which was shown in studies on challenges reported in SMEs reports . The challenges existed even before the implementation of MPs and were connected with infrastructure and management processes in SMEs. The challenges reported by studies in large organizations are mostly related to the issues discovered while implementing MPs.
Conclusion: The analysis of measurement models, tools, metrics selection methods and challenges of implementing MPs should help SMEs to make a feasibility study before implementing a MP.

Keywords: software measurement process, software measurement program, small and medium enterprise (SME), software metrics, software measures, systematic mapping study, GQM

1. Introduction

The number of Small and Medium Enterprises (SMEs) in the software industry is rising quickly and contributing significantly to the Gross Domestic Product (GDP) [1]. The definition of SMEs varies from country to country. According to the European Union [2], "SMEs are those companies which employ fewer than 250 employees and which have an annual income not more than 50 million euro, and/or an annual balance sheet total not more than 43 million euro" [3]. The firms which employ fewer than 50 employees are known as small enterprises and the firms which employ a maximum of ten or in some situations five workers are known as micro-enterprises [3]. SMEs play a very important role in supporting the economy and growth of any country [4].

The software development organizations, just like any other organization, aim to deliver products and services with expected quality by effectively using resources within software development processes. Software measurement is essential to characterize, evaluate, predict and improve

software products, processes and resources. Every software development process either generates or uses measurement data. The software measurement domain presents various measurement models, tools and practices to collect and analyse measurement data to estimate, monitor, control and improve software processes, products and resources. Software development organizations implement measurement programs (MPs) as part of software measurement process [5].

It is discussed in a recent SLR [6] that most of the MPs in large organizations fail to achieve measurement objectives and usually they do not sustain more than two years due to multiple reasons [6]. The rate of failure in the successful implementation of MPs is particularly exceptional in the perspective of SMEs [7, 8]. The MPs at SMEs become challenging because they usually do not have enough time, budget and resources to implement measurement plans. Software measurement knowledge is particularly poor in SMEs [7, 8]. The use of software measurement is limited in SMEs due to the lack of metric selection methods [9], a different set of metrics used in different SMEs [10], the lack of infrastructural facilities, low measurement maturity level, small development teams, higher workload [11] and limited measurement planning [12, 13].

A comprehensive Systematic Literature Review was conducted on software MPs and it was observed that there were fewer than 10 percent primary studies on implementing MPs in SMEs [6]. Therefore, this study presents a Systematic Mapping Study (SMS), which specifically focuses on measurement models, tools, metrics selection methods, and challenges of implementation software MPs at SMEs. Later, the measurement models, tools, metrics selection methods, and challenges in the implementation of MPs in SMEs and large organizations [6] are also compared. The measurement studies are analysed by answering the following research questions (RQs). There is no such study published with research questions (presented below) to the best of our knowledge.

RQ1: What measurement models, tools and practices for implementing measurement programs in SMEs are discussed in literature?

RQ2: What are the problems, challenges and issues of implementing measurement programs in SMEs?

RQ3: What metrics selection techniques, methods and approaches are used for measurement programs in SMEs?

This paper is organized as follows: Section 2 presents related work, Section 3 presents Systematic Mapping Process, Section 4 presents results and analysis and Section 5 presents conclusions.

2. Related work

Kitchenham [14] conducted a mapping study to investigate the status of software measurement research between 2000 and 2005. She identified that software MPs were the most researched area of the software measurement domain [14]. The journal papers were found to be more influential in measurement community than conference papers based on the numbers of citations. The study concluded that the there is a need for comparative studies and to serve this purpose empirical datasets should be made public. The datasets used among the primary studies were categorized as public (31%), private (61%), partial (8%) and unknown (1%). The primary studies lack the discussion on lightweight measurement methods for SMEs.

Gómez et al. [15] conducted an SLR to answer fundamental questions of what, how and when to measure. They analysed 78 primary studies. The measurement aspects discussed among the primary studies were categorized as project, process and product. They established that most of the primary studies discussed product metrics (79%) followed by project (12%) and process (9%) metrics. The software complexity and its size were identified as the most frequently measured attributes. The software metrics were mapped to typical initial, intermediate and final phases of a software project life cycle. Most of the metrics were found to be utilized for the initial phase (48%), followed by the intermediate (36%) and final (16%) phase. They concluded that software metrics need theoretical and empirical validation before being used in a measurement process. The

discussion and primary studies on lightweight measurement methods and measurements used in SMEs are missing in the SLR.

The software measurement process has a key objective of predicting the use of measurement data and software defects as they are one of the most predicted attributes [6, 14]. Catal et al. [16] conducted an SLR to analyse the software defects prediction studies. They analysed 75 primary studies published between 1990 and 2009 and classified the primary studies according to methods used for fault prediction, i.e. machine learning methods/algorithms, statistical and machine learning methods and expert judgment. The machine learning and statistics are found to be the most widely used methods for software measurement. Furthermore, fault prediction metrics were classified with respect to method, class, component, file, process and quantitative-values levels. They found out that 60% of studies used method-level metrics and 24% of studies applied class-level metrics and only 4% of studies have used process-level metrics.

Malhotra [17] conducted an SLR on software defect prediction studies published between 1991 and 2013. They found that most of the studies use size, effort and object oriented metrics for prediction. Radjenović et al. [18] conducted an SLR on fault prediction studies which were published between 1991 and 2011. They identified that object-oriented metrics (49%), traditional source code metrics (27%) and process metrics (24%) are mostly used in fault prediction studies. They found out that defect prediction studies mostly used one type of metrics, e.g. method-level, class-level, process-level, or source code metrics or object-oriented metrics. Hall et al. [19] conducted an SLR to analyse 208 fault prediction studies that were published between 2000 and 2010. They established that studies which used a combined approach (where more than one type of metrics were used) performed better than the studies which used a single type of metrics. They found that the machine learning methods were mostly discussed. These methods focused on utilizing large amounts of data. They observed that the machine learning methods outperform the statistical methods because they overcome the shortcomings of traditional statistical processes. The discussion on lightweight prediction methods, which consider the minimal budget, time and resources of SMEs, are currently missing from the discussed SLRs.

Unterkalmsteiner et al. [20] conducted an SLR to analyse measurements and evaluation strategies, which are used to assess the software process improvement (SPI) initiatives. They analysed 148 primary studies that were published between 1991 and 2008. The studies were classified with respect to their measurement focus, process quality, and prediction/estimation accuracy and software measurements (such as size, effort and customer satisfaction). The SPI models are discussed and the capability maturity model (CMMI) is identified as the most studied model in the SPI domain. The primary studies mainly focused on the measurement of quality (39%), prediction accuracy (38%) and productivity (35%). Three levels of measurements are explored, i.e. product, project and organization. The measurement of SPI initiatives is mostly done at project and project-product level. The problems in SPI studies are discussed, e.g. more than half of the studies do not completely describe the SPI context (organizational size, measurements validity and scope of SPI activities, etc.). They considered that the lack of context description might hinder the reuse of learned lessons and results in similar settings. This study is different from this research in two ways; 1) it does not discuss the role of MP for SPI and 2) it mainly focuses on SPI for large companies as there is no discussion and paper found on SMEs.

Touseef et al. [6] conducted an SLR on software MPs by analysing 65 primary studies that were published between 1997 and 2014. They analysed 35 measurement planning models, 11 associated tools, and metrics selection methods, and success/failure factors for implementing MPs. Most of the models and tools extended goal-based measurement approaches. The measurement studies are categorized with respect to measurement purposes, i.e. characterization (81%), evaluation (77%), prediction (28%) and improvement (70%). The measurement planning models and tools are categorized based on mea-

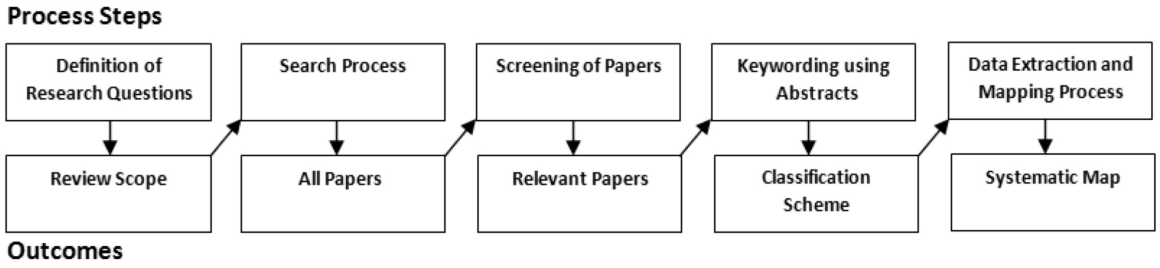


Figure 1. The systematic mapping study (SMS) process [21]

surement entities, i.e. processes (96%), products (58%) and resources (40%). The success factors for implementing MPs include organizational adoption of an MP, and Integration of an MP with SDLC, the synchronization of an MP with an SPI. Most of the measurements planning models were evaluated in case studies. They found that there are few measurement studies with the comparisons and reusability of results and the learned lessons of implementing MPs. The lack of context description (e.g. organizational size, measurement scope, and measurement analysis methods) hinders the reusability and comparative analysis of results among primary studies. The metrics datasets used in MPs are not explicitly presented in the measurement studies. In this study, only 3% of the studies discuss measurement planning models and tools for SMEs. Therefore, this SMS was conducted to specifically analyse measurement models, tools, and metrics selection methods that are proposed for SMEs while considering specific challenges in the implementation of MPs in SMEs.

Sulayman et al. [22] conducted an SLR on software process improvement (SPI) in small and medium web companies. The aim of the study was to specifically identify SPI models and techniques for small and medium web companies. They analysed only 4 primary studies after applying inclusion/exclusion criteria based on research questions. They found the limitations of SMEs, such as tight budget, ambitious deadlines and short-term strategy. The success factors include an increase in productivity, compliance with standards and overall operational efficiency. Pino et al. [23] conducted an SLR to analyse SPI approaches in SMEs by analysing 45 primary studies published between 1996 and 2006. They found CMM (38%) as the most discussed SPI standards in primary studies. They found that other standards, such as ASSESSMENT SEI (16%), IDEAL (13%), CMMI (9%), SPICE (13%), ISO/IEC 12207 (11%), GQM (2%) and PSM (2%), are not frequently used in SMEs. They also established that SPI is mostly measured in terms of employee perception instead of a formal measurement process. They claimed that the most frequently used SPI model for SMEs is CMM used as a reference model, ISO 15504 as a process assessment model and the IDEAL model for guiding improvement. It was also established that SMEs found it hard to implement SEI and ISO models. The RQs answered in these studies ([22, 23]) do not discuss the role of MPs for SPI, but rather the role of measurement for SPI. This study provides an analysis of the implementation of MPs in SMEs with respect to characterization, evaluation, improvement and prediction.

3. Systematic mapping process

This section presents the planning of Systematic Mapping Study (SMS) to analyse the existing literature regarding MPs at SMEs [21]. The overall steps of an SMS process are presented in Figure 1. The goal of this SMS is to systematically recognize, explore, and classify the studies on software MPs at SMEs and present the mapping of these MPs to highlight their possible challenges and the future scope of study [24]. The SMS was performed following the guidelines in [25] and implemented the systematic mapping process proposed by Petersen et al. [21]. Each step of the SMS process has an outcome and the overall outcome of the process is a systematic map.

Table 1. Research questions of systematic mapping study

ID	Research question	Motivation
RQ1	What measurement models, tools and practices for implementing measurement programs in SMEs are discussed in literature?	To understand the reported measurement models, tools and practices developed in SMEs to implement software measurement programs.
RQ2	What are the problems, challenges and issues of implementing measurement programs in SMEs?	To understand problems, limitations and challenges faced by SMEs during the implementation of measurement programs.
RQ3	What metrics selection techniques, methods and approaches are used for measurement programs in SMEs?	To highlight the metric selection methods used in different SMEs for implementing their measurement programs.

Table 2. Search string

Population	Intervention
(software) AND ("measurement program" OR "measurement process") AND "small and medium enterprise" OR SME)	(metric∗ OR measur∗ OR model OR framework OR tool OR challeng∗ OR problem OR issue OR improv∗ OR goal)

3.1. Definition of research questions

The main objective of this mapping study is to determine how software MPs are implemented in small and medium enterprises (SMEs). To answer this question, three research questions (RQ) were defined, as presented in Table 1.

3.2. Search process

A search string is used to select a potentially relevant set of primary studies. The lack of consistency for measurement concepts and terminology is a major threat to finding the relevant studies [26]. Therefore, initially the main concepts and terminology in the software measurement domain were reviewed and then the keywords considering the RQs were identified. Then, the synonyms and alternatives for each keyword were checked. Finally, "AND" and "OR" operators and wildcard character "∗" were used to create the search string. The "OR" operators were used to combine synonyms. The wildcard character "∗" was used to represents zero, one, or multiple alphanumeric characters in the position it occupies. The "AND" operator was used to combine the search string between population and intervention as shown in Table 2.

Population: In software engineering, population may refer to a particular software engineering role, the category of software engineer, an application area or an industry group [27]. In our perspective, the population is (software) AND ("measurement program" OR "measurement process") AND ("small and medium enterprise" OR SME). In population, the keyword "Software" is used to search studies related to software engineering only. The keywords "measurement program" and "measurement process" are used to search studies which discuss a measurement program or a measurement processes. The keyword "small and medium enterprise" and SME cover small and medium enterprises.

Intervention: In software engineering, intervention refers to a software methodology, tool, technology, or procedure. In this case the intervention is clear according to the situation of this study, that is (metric∗ OR measur∗ OR model OR framework OR tool OR challeng∗ OR problem OR issue OR improv∗ OR goal). The keywords "metric" and "measur" refer to the metric/metrics and measure/measures/measuring/measurement, respectively. The keyword "improv∗" refers to the variations of improve such as improv-

ing/improves/improved. The "challeng" refers to the variations of challenge such as challenges/challenged/challenging.

The primary studies were selected by reviewing the titles, abstracts and conclusions of the search results obtained from different databases. The databases were selected based on the experience reported by [6]. Table 3 presents the number of search results per research database.

3.3. Screening of relevant papers

This step of SMS is completed by applying study inclusion and exclusion criteria.

Study exclusion criteria:

The studies which do not conform with the exclusion criteria were excluded:

- studies which are not reported in the English language;
- studies which are not accessible in full-text;
- books and grey-literature;
- studies conducted in non-software companies.

Study inclusion criteria:

General criteria:

- a study is conducted in SMEs context;
- a study is in the area of software metrics and software measurement programs/ processes;
- a study includes an empirical evaluation (experiment, case study, survey, experience report, and/or action research).

Criteria specific to research questions:

- a study presents discussion/analysis on software measurement models or tools in SMEs (RQ1);
- a study discusses challenges, issues, limitations and problems that are related to software measurements in SMEs (RQ2);
- a study discusses metric selection methods for implementing software measurement programs in SMEs (RQ3).

Figure 2 presents the selection of the final set of primary studies (35) after applying the search process, exclusion/inclusion criteria, and snowball tracking. The snowball tracking reviews the references of every primary study with respect to its relevance to research questions. Endnote, a reference management tool, is used to remove duplicates and to manage the large number of references.

3.4. Keywording

The objective of keywording is to effectively produce a classification schema and ensure that all the selected papers are relevant [21]. Figure 3 shows the systematic process that was followed to create the classification schema.

The initial step comprised reviewing the abstracts of primary studies and then allocating them a number of keywords to recognize the basic contribution topic of the article. After that, all the keywords were consolidated to establish the high-level of classification, and to understand the area of research highlighted in the primary studies. The schema experienced a continuous improvement process by logically fitting the papers into classes for new data. The resulting classification schema is presented below.

The primary studies are classified based on the following schemas:

- **Time of publication:** to map the studies based on the time of publication.
- **Empirical research method:** to map the study according to the research method used.
- **Contribution type:** to map the outcomes of different types of studies.
  - **Models/tools:** to map the models, tools, and measurement methods for building software measurement processes in SMEs.
  - **Challenges:** to map the studies, which discussed challenges, issues, limitations regarding software measurements in SMEs.
  - **Metric selection criteria:** to map the studies which discussed metrics selection methods and most commonly collected metrics in SMEs.

The *time of publication* schema describes the number of primary studies which are related to research questions.

The *empirical research method* is the classification schema which categorizes the studies based on their research methods as presented in Table 4. The research methods are categorized as a case study, survey, industrial report and experiment.

The *contribution type* schema describes the type of contribution by study. It is classified into models/tools, measurement methods in SMEs, metric selection methods, com-

Table 3. Number of studies retrieved per research database

Research resources used	Number of potential primary studies
Search Engines	
Google Scholar	1960
Wiley Interscience	34
Science Direct Journals	06
Springer	117
One Search (Search Tool)	2372
ACM	50
IEEE Xplore	99
Journal Databases	
ACM Transactions on Software Engineering Methodology (TOSEM)	10
IEEE Transactions on Software Engineering (TSE)	2
IEEE Software	4
Software Quality Journal	3
Journal of Systems and Software	1
Empirical Software Engineering	38
Automated Software Engineering	0
Conference Databases	
IEEE International Software Metrics Symposium (2000-2005)	3
IEEE International Conference on Software Engineering (ICSE)	0
Joint International Conference on Software Process and Product Measurement (Mensura) and Workshop on Software Measurement (IWSM)	5
Empirical Software Engineering and Measurement (ESEM) (2007-2014)	0
Product Focused Software Process Improvement (PROFES)	11
Software Process and Product Measurement	0
Software Engineering and Advanced Applications (SEAA)	3
Pacific Industrial Engineering and Management Systems (APIEMS)	4
European conference on software process improvement (EuroSPI)	5
International Conference on empirical Assessment in Software Engineering (EASE)	0
International Symposium on Empirical Software Engineering and Measurement (ESEM)	0
Information Technology: New Generations (ITNG)	0
International Conference on Emerging Technologies (ICET)	1
Total	**4728**

monly selected metrics and challenges related to the implementation of MPs in SMEs. The Model/tools are further categorized into extended goal question metric (GQM) methodology or software process improvements (SPI) methodology or measurement process improvement.

The metric selection criteria are also categorized into three subclasses; use of standards, use of measurement expert and experience and use of automated tools. These three subclasses were earlier defined based on the analysis of metrics selection methods used in the measurement studies in [6]. The mapping results of the classification schema are analysed in Section 4.

3.5. Data extraction and mapping

A data extraction form was developed in MS Excel (Table 5) to extract data from the primary studies for each RQ using the classification schema.

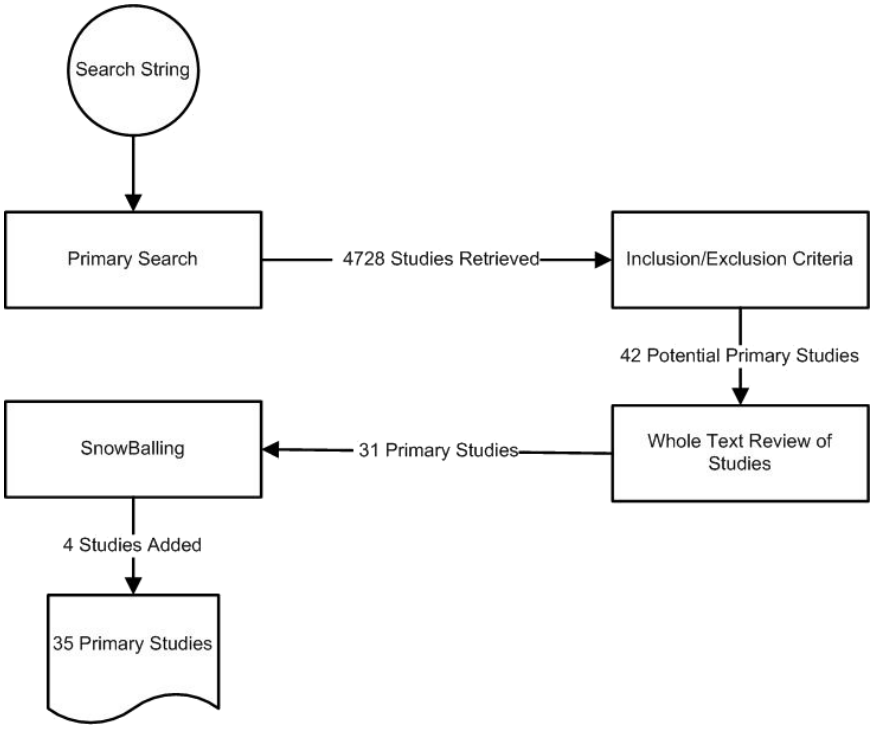


Figure 2. Process of selecting primary studies

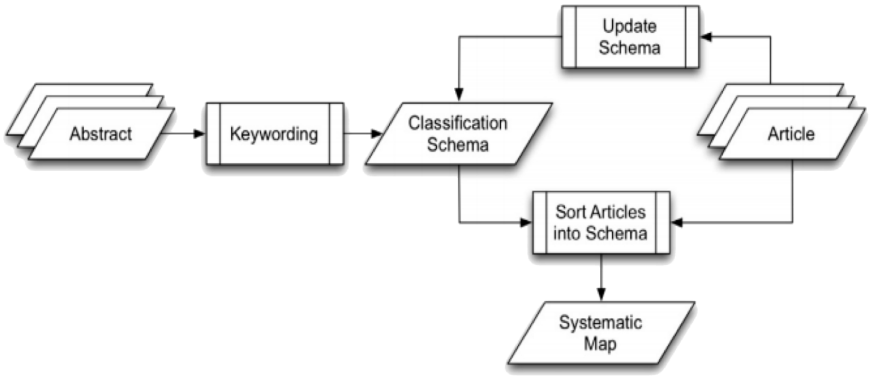


Figure 3. Creating the classification schema [21]

4. Results and analysis

In total, 35 measurement studies are analysed in this section. First a short overview of the studies is presented with respect to publication year and research method. It is followed by the presentation of results and analysis.

Publication year: The results of the systematic mapping study are presented in this section. In total, 35 primary studies are analysed and Figure 4 presents the numbers of primary studies with respect to the year of publication. The numbers of primary studies on implementing MPs in SMEs are smaller as compared to 65 primary studies on implementing MPs in large organizations in our previous study [6]. Therefore, it is important to discuss the history of software measurement domain and how it became critical of SMEs.

Software measurement is a young discipline as the history of software metrics dates back to the late 1960s [31]. It is claimed in [31] that the first book on software measurement [32] was published in 1976 and the first comprehensive report on implementing software MPs was published by Grady and Caswell [33] in 1988. The widely

Table 4. Classification schema of research methods

Purpose	Meta-data
Survey	A research method designed and performed to observe the opinions of people in a structured way [28]
Case study	A research method considered and presented to examine the opinions of people in an unstructured way [28, 29]
Experiment	A research method designed and performed to work with one or more variables and manage all other variables to measure results [30]
Industrial report	A research method used to evaluate the industrial experiences without clear research questions and objectives [30]

Table 5. Data extraction form

Purpose	Meta-data
General information	Study title, authors' names, date of publication and research methodology
Specific information	Measurement models/tools at SMEs, metric selection methods, commonly selected metrics and challenges/problems/limitations in the implementation of measurement programs in SMEs

used Goal Question metrics (GQM) model [34] was also introduced in 1988 and the first comprehensive guidebook on goal-oriented measurement was published by Park in 1996 [35]. Software MPs in large organizations have faced many challenges over the last three decades [6, 36].

The evolution of software engineering and software industry includes interdependencies and has impact on the emergence of SMEs. The SMEs started to influence the software development industry following the advancements in microchip technology and communication technologies (e.g. the internet) and the unbundling of software from hardware. According to [37], internet services also affected SMEs based on four factors. The first factor is access to global information sources to enable extension in a business network. The second factor is enabling faster document transfer, online transactions and faster communication channels. The third factor is enabling the search of low cost market, minimizing dependency on a local market (e.g. outsourcing, crowd sourcing and global software engineering). The fourth factor is feedback by international clients and adapting globally successful strategies.

Researchers and practitioners specifically aimed to design software development processes for SMEs during the mid-1990s. There is a plethora of studies published between 1995 and 2000 to promote iterative and incremental software development for the different structure and limitation of SMEs [38]. Basili and Larman claimed in their book ([38]) that the first book on agile software development (e.g. SCRUM, XP) was published by Cockburn [39] in 2002. SMEs represent 99 percent of businesses in Europe[1] with respect to the currently used definition of SMEs that was legislated in 2003. This definition is an updated version of the 1996 definition.

It might be argued based on the above discussion that software engineering research community initially focused on software development processes (e.g. Waterfall, Spiral) and software measurement processes in large companies. Later, the research community focused on software development processes (e.g. Agile, SCRUM) in SMEs when these processes became operational and popular, then they specifically focused on software measurement processes for the characterization, evaluation, prediction and improvement of software development processes in SMEs.

The first study meeting the inclusion criteria was published in 2001. Therefore, this paper presents the search period between 2001 and

[1] http://ec.europa.eu/growth/smes/business-friendly-environment/sme-definition_en

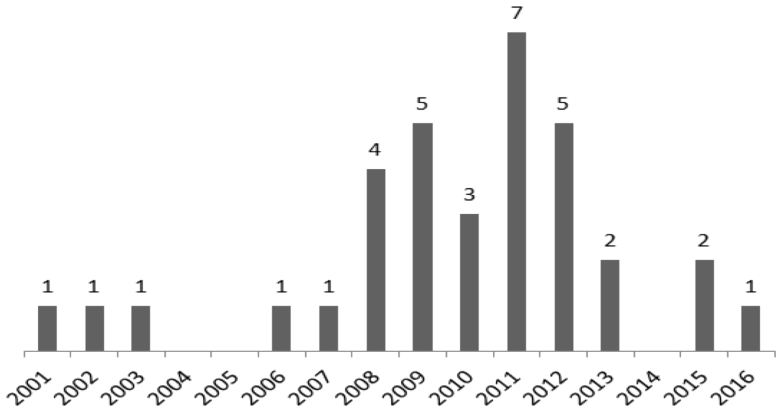


Figure 4. Distribution of primary studies with respect to time of publication

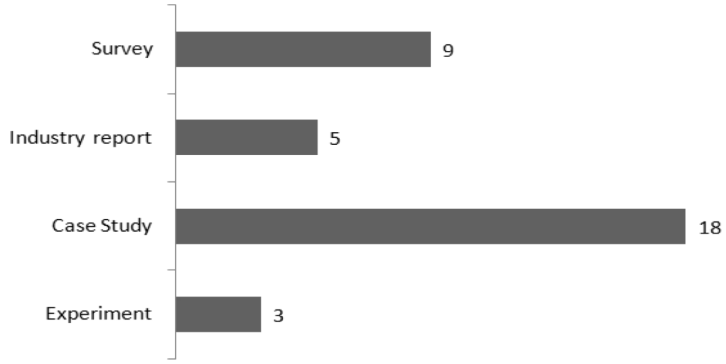


Figure 5. Distribution of primary studies with respect to research methods

2017 in Figure 4. The research databases shown in Table 3.

Research method: The most commonly used research methods in selected studies are case studies (51%) and surveys (25%). Some of the studies used industrial reports (14%) and experiments (8%) as shown in Figure 5.

RQ1: "What measurement models, tools and practices for implementing measurement programs in SMEs are discussed in literature?"

Touseef et al. [6] conducted an SLR on software MPs by analysing 65 primary studies, they studied 35 measurement planning models. In their study [6], they found only 4 specifically defined measurement models for SMEs. They observed that 83% (29 out of 35) measurement models extended the goal-oriented approach or the goal question metric model. The concept behind goal-based approaches is to identify the measurement goals of an organization and then the relevant metrics to achieve measurement goals [34, 35]. In this SMS, 29 software measurement models and 4 tools among 35 primary studies were identified.

Table 6 presents the "Base Measurement Model", of the "Measurement Model" and its "Measurement Purpose" and "Implementation Purpose". The "base measurement model" in Table 6 refers to the parent model of the identified "Measurement model" for SMEs. The "implementation level" refers to the implementation levels of MPs (i.e. project level and/or organization level). The "measurement purpose" represents the basic purpose/objective of MPs discussed in the studies (i.e. to Evaluate (E), Improve (I), Characterize (C) and/or Predict (P) the software process, product or resource entities) [34, 35].

Figure 6 presents the categorization of measurement models. These models are categorized among "goal oriented approach improvement (GOAI)", "software process improvement (SPI)" and "measurement process improvement (MPI)".

The PRISMS model is based on the goal-oriented measurement and SPI. Similarly,

Table 6. Software measurement models for SMEs

ID	Base	Measurement model	Implementation level	Measurement purpose
S16	GQM	Light weight GQM	Organization	CEI
S2	GQIM, CMM	MIS-PyME MCMM	Project	CEIP
S1	GQM, GQIM	MIS-PyME	Project	CEIP
S5	GQM, GQIM	MIS-PyME	Organization	CE
S3	GQIM	MIS-PyME methodology	Project, Organization	CEI
S4	GQIM	MIS-PyME	Organization	CEI
S6	CMMI 1.2	SQIP	Project	EI
S8	GQM, CMM	PRISMS	Project	CEI
S9	CMM	MESOPYME	Project, Organization	I
S10	QFD	SPM	Organization	EI
S11	CMMI	AAHA	Organization	I
S12	TQM	LQIM	Organization	EI
S14	BSC	HSC (Holistic Scorecard)	Organization	EI
S15	No Base Model	Pro Scrum	Project	I
S20	GQM	GQM-DSFMS	Project	CEI
S19	No Base Model	Tarc	Project	C
S21	GQM	Four step framework	Organization	CI
S22	GQM	OMSD	Project	CEI
S23	GQM	SPGQM	Project	CEI
S24	No Base Model	SCAPT	Organization	CEI
S26	QIP, SME	AM-QuICk	Project, Organization	EI
S27	CMMI, PSP, XP, SCRUM	ASPISME	Project, Organization	EI
S28	ISO/IEC 12207:2008, SCRUM	Adapting ISO/IEC 12207:2008 for SCRUM	Project, Organization	CEIP
S29	SWEBOK	Adapting ISO/IEC 15939:2007	Project, Organization	CEIP
S30	ISO/IEC 15504, ISO/IEC 12207:2008 and CMMI	Hybrid Process Model	Project, Organization	CEIP
S31	GQM	GQM Adaption for SPI	Project, Organization	EI
S32	ISO/IEC 12207:2008	COMPETISOFT	Project, Organization	CEP
S33	No Base Model	PMS-IRIS	Project, Organization	EI
S35	CMMI, SCRUM	CMMIbyScrum	Project, Organization	CEI

MIS-PyME, MCMM, and 4-step framework extend goal-oriented measurement and MPI. The AAHA model is proposed to enable SPI and MPI in SMEs. An interesting finding is that the numbers of SMEs are increasing rapidly throughout the world but there are limited numbers of studies that present measurement models/tools for small and medium enterprises as compared to large organizations [6]. For instance, SMEs represent 99 percent of businesses in Europe[2] with respect to the currently used definition of SMEs that was legislated in 2003. SMEs face challenges such as having limited resources, shorter time to market, limited budget, and frequent changes in customer requirements [S1, S2, S3, S4, S5]. Therefore, there is a need for specific models/tools to deal with particular challenges to the establishment of MPs in SMEs. Pino et al. stated in an SLR [23] that ISO and SEI standards for SPI are not directly suitable for SMEs due to the com-

[2]http://ec.europa.eu/growth/smes/business-friendly-environment/sme-definition_en

plexity of recommendations and the requirement of large investment of time and resources. Therefore, there is need for widely accepted strategies to adapt these standards in SMEs [23]. It was proposed to adapt the guidelines and methods used in the measurement models that are already reported for large organizations in this SMS [6, 20]. The MIS-PyME [S2], SQIP [S6], PRISMS [S8], MESOPYME [S9], AAHA [S11], ASPISME [S27], and CMMIbyScrum [S35] models are proposed for the CMMI standard in SMEs. Irrazabal et al. proposed guidelines to adapt ISO/IEC 12207:2008 standard to SCRUM [S28] and María et al. proposed guidelines to adapt ISO/IEC 15939:2007, ISO/IEC 12207:2008 and CMMI in SMEs [S29].

Goal-oriented approach improvement (GOAI): In total, 29 measurement models are identified in this SMS and 40 percent of these models are proposed as the extension of goal-oriented approaches. For example, lightweight GQM process [S16] is an enhancement of the GQM model that is proposed to decrease measurement overhead considering the characteristics and limitations of small software companies. The OMSD [S22] model is proposed to select the optimum number of measures from the available large set of measurements within limited time and effort using meta-measures, such as collection time, cost, priority, value, and usage. The GQM model lacks a method to define measurement goals and questions in a consistent, complete, traceable and verifiable way [6]. Therefore, the SPGQM [S23] model extended the GQM model to define measurement goals and questions in a consistent, complete, verifiable and traceable way. The SPGQM model also used the OMSD model for the optimum number of metrics selection in a case study. GQM-DSFMS [S20] extended the GQM model to select the optimum number of metrics based on time, the cost and usage of metrics and the importance of measurement goal. It also presented a method to enable traceability among measurement goals, questions and metrics. Jezreel et al. [S31] proposed a method for applying the GQM model in SPI by conducting structured interviews of top management and operational management to define measurement goals, and then identify questions and metrics to achieve the goals. Similarly, the PRISMS model [S8] is proposed to relate business goals and improvement goals with measurement goals. Furthermore, the CMM model is used as a reference model to plan and implement MPs in SMEs. The MIS-PyME MCMM model [S2] is proposed to define the SMEs version of the CMM standard for SPI using the goal-oriented approach. The MIS-PyME model and its extensions are proposed with case studies to implement goal-oriented measurement processes and measurement process improvement in SMEs [S1, S3, S5].

Measurement process improvement (MPI): In total, 13 models are developed for improvements in measurement processes in SMEs. For example, the MIS-PyME [S1, S3, S4, S5] framework is presented to define the software MPs in SMEs. This model extended GQM and GIQM [40] to implement and improve the measurement process in small organizations. The MIS-PyME measurement capability maturity model [S2] was developed to support SMEs in defining MPs with respect to measurement maturity of the company and establishing a mechanism for the continuous improvement of MPs.

The LQIM [S12] model is presented based on the Total Quality Management (TQM) paradigm [41] to implement quality improvement plans in SMEs in Pakistan. It is recommended to use it with Deming"s Plan, i.e. Plan, Do, Act, Check (PDAC) for continuous improvement in quality processes. Caballero et al. [S15] present industrial experience related to MPI using agile methodology in SMEs. The study showed that Scrum might improve productivity without decreasing product quality in SMEs. The study [S15] also showed that Scrum is a good alternative for process improvement in an organization with very limited resources. A "four step framework" [S21] was presented to implement MPI in those SMEs which needed improvement in their development processes.

There are four measurement models proposed with the intentions of SPI and MPI simultaneously. AAHA [S11] is a lightweight method developed for SPI in SMEs, it is based on CMMI,

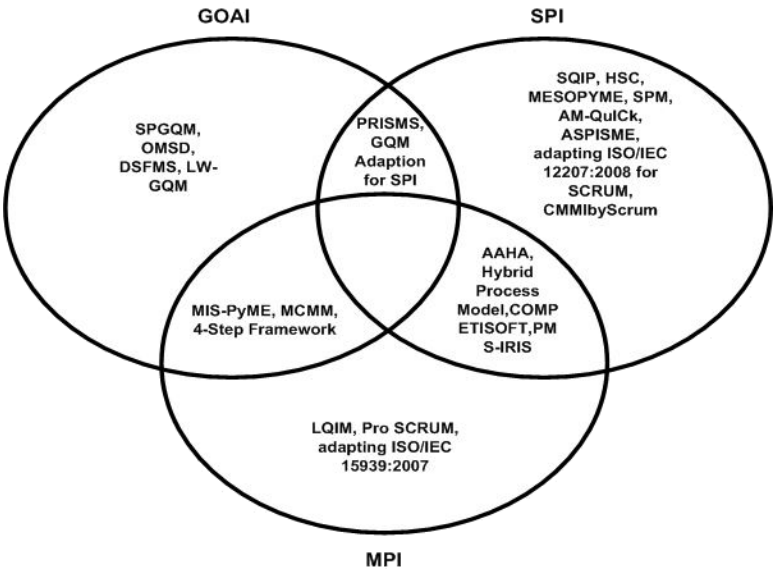


Figure 6. Categorization of measurement models with respect to goal-oriented approach (GOAI), software process improvement (SPI) and measurement process improvement (MPI)

SPICE and agile practices. It is particularly developed to provide a low cost improvement in the software development practices in SMEs. The Hybrid measurement model [S30] is proposed to adapt ISO/IEC 15504, ISO/IEC 12207:2008 and CMMI Dev 1.3 for the maturity of a measurement process and improvement in agile processes in an organization.

The COMPETISOFT model [S32] is based on the experience of using ISO/IEC 15504 and ISO/IEC 12207:2008 in 20 SMEs. It defines four steps of planning SPI, i.e. SPI definition, assessment, measurement and establishment. The improvement of documentation and project management processes is identified as the focus of most SPI initiatives in 20 companies. The PMS-IRS model [S33] proposed 9 steps of performance measurement systems in SMEs, i.e. planning the project, definition of enterprise environment, designing key improvement processes, analysis and design process, definition of measurement process levels, validation of measurements, establishing technological infrastructure, and human resource management. It defines the performance management system as a set of dynamic and integrated metrics for the measurement and evaluation of business operations enabling decision making for SPI.

There are ten key process areas and 3 themes (measurement, quality and tools) of Software Engineering Body of Knowledge (SWEBOK) [42]. Abran et al. proposed extensions in the measurement process of SWEBOK [43]. Maria et al. [S29], further extended Abran's proposal to adapt it for SMEs. They extended the key process areas of measurement by defining new measurement processes for SME, i.e. "process and business assessment", "perform measurement process", "and evaluate measurement" and "experience factor".

Software process improvement (SPI): Software Process Improvement (SPI) is a systematic approach to continuously increase the efficiency and effectiveness of processes in software development companies [20]. The SPI models proposed for establishing MPs in large organizations are not considered suitable for SMEs due to their complex nature and expensive cost [44]. SPI is one of many factors that can affect the success of software development organizations [S14]. There are multiple SPI models identified (e.g. CMMI, CMM, SPICE, PSP, TSP, Six-Sigma, QIP, TQM) in an SLR [20]. The CMM, Six-Sigma, and CMMI models are mostly discussed for implementing measurement processes in large organizations [20]. The ASPISME model [S27] is proposed to adapt CMMI and PSP for improving XP and SCRUM software development processes in SMEs. The ASPISME model contains guidelines for process improvement at

three levels, i.e. enabling individuals to understand and practice SPI activities and enabling SPI at the project level and organization level. Similarly, Irrazabal et al. proposed guidelines to adapt ISO/IEC 12207:2008 standard for SCRUM based on experience in 25 SMEs.

On the other hand, there are fewer SPI models available for SMEs and they are not widely used either. For example, the PRISMS model [S8] uses the GQM model for software process improvements. It also relates improvement goals to business goals which help to choose and prioritize key process areas for improvement. The SQIP model [S6] is proposed to improve the quality and reliability of a software development process to achieve the business goals in SMEs. Specific process improvement activities are used in this project, such as requirements and change management. SQIP adopted CMMI version 1.2 as the base model for the implementation and evaluation of software process improvement in SMEs.

The SPM [S10] model is based on QFD (quality function deployment) methodology. It is proposed to define SPI plans and estimate the effect of each SPI practice on a specific software process. The MESOPYME model is proposed to improve the quality and productivity of software development processes using action package concept (i.e. a method to help faster and inexpensive SPI program implementation in SMEs). The HSC model [S14] extended the BSC [45] model to observe business success in software development in SMEs by enabling synchronization between software development processes and business operations.

Ayed et al. [S26] proposed the AM-QuICK model for improvement in agile methodologies with the help of a measurement process. They proposed customization of agile methodologies for continuous SPI at multiple levels, i.e. organizational level, process management level and product management level.

Figure 7 presents the distribution of the measurement purposes of measurement models for implementing MPs (i.e. evaluation, improvement, characterization and prediction). Characterization means that an MP is implemented to collect the data about potential causes of a problem or understand the state of processes, products or resources (e.g. to understand the delays in product delivery, MP implementation can help to collect data about the number of bugs reported, the number of change requests by customer). Evaluation means that an MP is implemented to gauge and analyse the gap between the planned and actual state of processes, products and resources (e.g. to analyse the difference between estimated and actual effort). The prediction means that an MP is implemented to use historical data to make an estimation about software processes, product and resources (e.g. to predict number of bugs in a software product). The improvement means taking actions to improve software processes based on the measurement process. The distribution of software MPs with respect to their measurement purpose are: improvement (86%), evaluation (80%), characterization (60%), and prediction (20%). When a combination of purposes (i.e. when more than one purpose was mentioned by a primary study) was investigated, it was found out that around 59% of the studies mentioned the purposes of characterization and improvement while only 17% listed all four purposes.

Figure 8 presents the distribution of the implementation levels of measurement models for implementing MPs (i.e. project and/or organization level). It was observed that most of the MPs are implemented at the organization level (45%) and the project level (45%) and only 10% of MPs are implemented at both project and organization levels.

RQ2: "What are the problems, challenges and issues of implementing measurement programs in SMEs?"

Table 7 presents the challenges of implementing MPs in SMEs.

Low measurement maturity: The implementation of software measurements processes in SMEs is limited due to low measurement maturity [S2]. It is stated in [S1, S2, S3, S4, S8, S25, S29, S30, S31, S32, S33, S35] that measurement processes are either not defined at all or poorly defined in SMEs, which hinders defining measurement indicators and measurement goals in

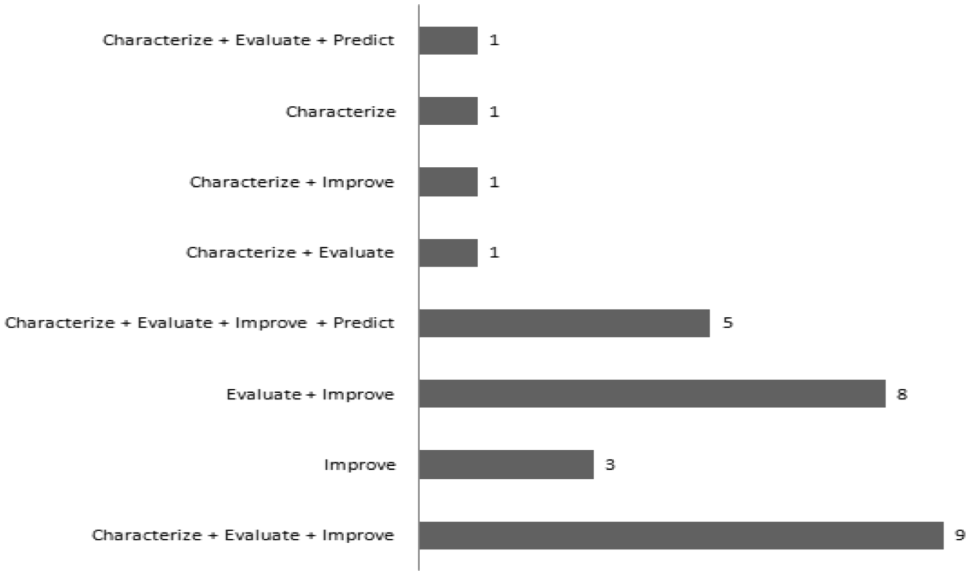


Figure 7. Distribution of measurement purposes of measurement models for implementing MPs

Figure 8. Distribution of implementation levels of measurement models for implementing MPs

SMEs. The SMEs do not have enough resources to promote serious MPI plans [S2, S9], [1, 10, 11]. All staff members are involved in the activities related to managing daily work and have no extra time for additional activities, such as implementing MPs. The implementation of MPs face major challenges such as limited resources to perform MPI [S9, S29, S33, S34, S35] and the lack of measurement experts [S12, S30, S32, S33, S35] and the lack of time for accurate estimations [S13, S29].

Poor software measurement knowledge: SMEs have poor measurement culture due to the lack of measurement knowledge, training and the perceived importance by administrators in SMEs [S12, S30, S32, S33, S35]. Therefore, a few measures are collected in these companies [S2, S21]. The lack of knowledge of measurement techniques among the software developers [S17], [11] also hinders the collection of measurement data. Developers seem to be in a great confusion about what to measure and how to measure [S17], [11]. They feel threatened by the possible adoption of a metrics program, as they perceive it as a tool that would be used for assessing their performance. Most of the developers have an insufficient knowledge of tools widely discussed and available in the literature. The management at SMEs usually do not understand the importance of a measurement process and the developers are mostly fresh university graduates equipped with insufficient knowledge about software quality and the importance of measurement [10, 46]. The people that are involved in MPs are not willing to use measurements due to their lack of knowledge of measurement techniques [S2].

Lack of experienced professionals: The stakeholders of the MPs including the measurement analyst usually come from the company implementing MPs. They usually have limited

Table 7. Challenges of implementing measurement programs at SMEs

Study ID	Challenges
S1, S2, S3, S29, S30, S33	Lack of measurement maturity for implementing software MPs Lack of experience in using data collection tools
S4, S29, S30, S31, S32, S33, S35	Lack of measurement maturity
S5, S25, S29	Scope of databases containing indicators and measures is small
S6, S29, S33	Formal process management techniques
S8, S25, S29, S33, S35	Lack of measurement maturity Lack of automated tool for data collection
S9, S29, S33, S34, S35	Limited resources to perform measurement improvements
S11, S33	Lack of formal measurement approach for software process assessment Software process assessment is time consuming and costly at SMEs
S12, S30, S32, S33, S35	Lack of measurement experts
S13, S29	Lack of time for accurate estimation of projects
S15, S30	Selected metrics are not verified for implementing measurements at SMEs
S17, S25, S29, S33	Use of metrics is limited due to unawareness of software measurement techniques among the software developers Measurement is considered a long-term activity Short time-to-market Use of metrics is limited due to lack of experienced professionals Measurements are limited due to lack of knowledge of quality issues in development process
S18, S33	Selected metrics are not validated for measurement and evaluation of SPI
S19, S33	The absence of automated tool for data collection Projects have a limited budget for empirical data collection and analysis
S20	Cost management (time and resources needed for collection and analysis of metrics) Redundancy in metric selection process
S22, S23	Redundancy in metric collection High effort required for metrics selection and collection
S24, S29, S31, S33	Unavailability of the required assessment data to measure
S4, S25, S28, S29, S30, S33, S34, S35	Lack of sync between measurement process and software development life cycle
S25, S26, S27	Incorrect definition of measures
S32, S33, S34, S35	Lack of sync between business objectives/strategies and SPI

expertise in the measurement field [S1, S2]. The SMEs should hire experienced professionals in permanent positions to plan, organize, implement, evaluate and improve MPs [8, 47]. A few case studies (e.g. [S1, S2, S3, S29, S30, S33]) showed that all of the measurement processes proposed in measurement studies are not possible to implement yet due to poor measurement maturity, poor measurement knowledge, and the lack of experience in using data collection tools. The SMEs face difficulties in hiring experienced professionals, because the offered reward is limited. Once the developers gain some experience, they seem to be inclined to migrate to larger companies hoping for better career prospects [S12].

Time to market: The use of software metrics is limited in SMEs due to challenging time to market with tight timeframes [S17]. Software developers in SMEs are always found battling with time pressures [S13, S29]. Most of the SMEs are aware that software measures are useful for improving quality but they believe that it re-

quires more time to implement a MP in the workplace [11].

Lack of measurement planning: Most of the SMEs have poor strategic planning processes for implementing their MPs due to barriers such as unavailability of assessment data [S24, S29, S31, S33], rapid application development [S13, S29], lack of formal process management, measurement management techniques and unwillingness to share ideas with employees [S6, S11, S29, S33], [46, 48, 49]. The lack of measurement planning also hinders linking measurement processes with business objectives and SPI [S32, S33, S34, S35].

Lack of automated tool support: The automated tools used in SMEs can be different due to multiple reasons. They can be different based on the implementation levels of MPs (i.e. organizational and/or project level), types of software entities to be measured (processes, products and/or resources), type of software development life cycle (e.g. agile, rapid application development), measurement purpose (characterize, evaluate, predict and/or improve software entities) and the business goals of software organization. There is a lack of automated tools for implementing software MPs in SMEs [11, 46, 50] as there are only four tools reported among 35 primary studies in this SMS (i.e. Tarc [S9], SCAPT [S24], SonarQube [S25], SPIALS [S35]). There is an increasing need for well understood and affordable tools that can select required metrics to implement software MPs in SMEs [S8] [46]. The automated tools might also help to overcome budget limitations, time and measurement experts in SMEs [S12].

The databases in SMEs contain a small number of measures and indicators [S5, S25, S29]. The small scope of measurement databases might be due to the lack of synchronization between a measurement process and a software development life cycle [S4, S25, S28, S29, S30, S33, S34, S35]. The lack of automation and small scope of databases causes redundancy in metrics collection and high effort is required for metrics collection [S22, S23].

Data collection problem: The unavailability of the required assessment data [S19, S24] for measurement tools is a critical challenge. This problem might not only reduce the descriptive power of the tool but also reflect company's operational problems. The tools perform effectively if the company has defined data collection and storage procedures [S19, S24]. Furthermore, projects have limited budget for empirical data collection and analysis [S19, S33]. Therefore, there is a need for automated tools, which can help to reduce the overhead associated with data collection and processing to perform measurements in SMEs [S8]. The lack of budget, time and resources also hinders the quality assurance process for the data collection process [S15, S30] and the validation of metrics for their suitability for SPI improvement [S18, S19, S20, S25, S26, S27, S33].

It was not possible to find any solution to the problem of initiating the data collection process in this mapping study, however, the SLR [6] revealed that Iversen and Mattiassen [51] discussed experiences in establishing an MP with the help of incremental application of GQM and intelligent collection and analysis of data. Therefore, the automation of data collection process can be incrementally implemented. The first step may include the collection of data with manual entries into measurement repository using a tool. In the second step, data collection may also be automated. This requires the integration of the MP with the SDLC [S4, S25, S28, S29, S30, S33, S34, S35]. There are both open source and commercial tools to automate the data collection for SDLC processes [52]. The use of automated tools for characterization, evaluation, and prediction of software processes, products and resources becomes even more important in SMEs because there is a shortage of time, human and financial resources in SMEs.

RQ3: "What metrics selection techniques, methods and approaches are used for measurement programs in SMEs?"

Table 8 presents the most commonly used metrics based on their frequency of being discussed among the primary studies. The Software metrics/measurement-attributes/measures are identified, collected and analysed based on the definition of specific measurement objectives (e.g. defect prediction, size estimation).

Gómez et al. [53] identified in a SLR that complexity and size are most discussed metrics

Table 8. Types of metrics/measures in primary studies

Metric/Measurement-attribute/Measure	Definition	Selected studies	Frequency
Defects	Errors or failures in a software product.	S1, S2, S3, S4, S5, S6, S8, S12, S17, S26, S27, S31, S33, S35	14
Productivity	The speed of software production in terms of effort and time.	S1, S3, S4, S5, S17, S14, S15, S16, S22, S26, S28, S29, S30, S33, S34	15
Customer satisfaction	The expectation of customer about the performance of software product.	S1, S3, S4, S5, S7, S10, S12, S14, S24, S31, S33, S34, S35	13
Size	The size of the product in the form of functional points or LOC.	S2, S6, S13, S15, S21, S22, S27, S33	8
Duration	The time required to construct software product.	S1, S2, S3, S4, S5, S6, S7, S9, S22, S24, S26, S29, S33, S35	14
Effort	The human effort to develop a software product.	S1, S2, S3, S4, S5, S13, S15, S16, S21, S22, S26, S29, S35	13
Reliability	Number of error-free operations in a system under particular conditions.	S1, S3, S4, S5, S24, S31	6
Traceability	A measurement that counts the software requirements that are not traced to the system requirements.	S6, S8, S18, S31, S33	5
Cyclomatic complexity	A measurement that shows the complexity of software product.	S2, S6, S8, S26	4

Table 9. Metrics selection methods

Metrics selection methods	Studies	Frequency	Percentage
Use of standards	S2, S6, S11, S17, S19, S20, S22, S23, S28, S29, S30, S32, S33, S35	14	40%
Use of measurement expert and experiences	S1, S3, S4, S5, S7, S8, S9, S10, S12, S13, S15, S16, S18, S26, S27, S28, S31, S32, S33, S34, S35	21	60%
Use of automated tools	S2, S9, S16, S19, S21, S25, S33, S35	8	22%

among primary studies on software measurement process in software development life cycle. An SLR [6] allowed to establish that defect, productivity and size are the most discussed metrics in large organizations. On the other hand, productivity, defects, effort and customer satisfaction are the most discussed metrics among primary studies in this SMS. There is an increasing need for a well understood and managed software measurement model in SMEs, to select the correct, relevant, timely, verifiable, cost-effective and valuable set of metrics [54].

In our previous study [6], metrics selection methods are classified as (i) use of standards, (ii) use of measurement experts and experience and (iii) use of automated tools. The same classification was used for metrics selection methods in this SMS as shown in Table 9. In this SMS, the use of a measurement expert and experience is the most practiced method among primary studies.

Use of standards: In an SLR on MPs [6], the primary studies discussed the role of standards such as ISO/IEC 15939:2007 [55], ISO/IEC 25000 [56], ISO/IEC 9126-x [57], ISO/IEC 14598-x [58], ISO/IEC/IEEE 24765:2010 [59], CMMI [60, 61], ISO/IEC 25021 [62], and ISO 9126 standard family [63–65] for the implementation of MPs.

In another SLR [20], the primary studies discussed the role of SPI models (SPICE, PSP, TSP, Six-Sigma, QIP, TQM) [66] and standards (e.g. CMMI, CMM, ISO 15504 [53] and ISO 9001 [53]) for the implementation of MPs. On the other hand, Pino et al. in an SLR [23] considered that ISO and SEI standards for SPI are not directly suitable for SMEs due to the complexity of recommendations, and the requirement of a large investment of time and resources. Therefore, they considered a need for widely accepted strategies to adapt these standards in SMEs and organizations. Furthermore, they considered that organizations which develop international Software Engineering standards should separately consider the measurement processes of SMEs [23].

In this SMS, multiple studies (e.g. [S2, S6, S11, S17, S19, S20, S22, S23]) stated that measurement standards (e.g. ISO/IEC 15504 [53], ISO 9001 [67]) are used to select metrics in different SMEs. The primary studies proposed multiple models to adapt those measurement standards in SMEs which are reported for MPs in large organizations. The MIS-PyME [S2], SQIP [S6], PRISMS [S8], MESOPYME [S9], AAHA [S11], ASPISME [S27], and CMMIbyScrum [S35] models are proposed to adapt CMMI standard to SMEs. Irrazabal et al. proposed guidelines to adapt ISO/IEC 12207:2008 standard for SCRUM [S28]. Similarly, Marìa et al. proposed guidelines to adapt ISO/IEC 15939:2007, ISO/IEC 12207:2008 and CMMI in SMEs [S29].

In [S1, S2, S3, S4, S8, S25, S29, S30, S31, S32, S33, S35], there is a proposal to implement MPs in SMEs according to the maturity level of software processes in the company. The MIS-PyME measurement capability maturity model [S2] for implementing MPs in SMEs uses ISO/IEC 15504 standard as a reference model [53]. The SPI models use measurements as the key component of their processes. For instance, the CMMI model contains guidelines for defining the measurement process and then using this process to monitor and control software development processes. Later, the collected measurement data is used for quantitative management and continuous improvement.

In [S20, S22, S23], the idea of using a predefined pool of standard metrics is proposed. The software companies can choose metrics from this pool based on their measurement goals using meta-metrics (importance of metrics for measurement goal, cost/time of metrics collection, and frequency of metrics usage in measurement project). The usage of a common set of metrics for different projects which have similar goals, might reduce the effort and cost of data collection.

Use of measurement experts and experience: Most of the SMEs use measurement experts and experiences to select metrics [S1, S3, S4, S5, S7, S8, S9, S10, S12, S13, S15, S16, S18, S26, S27, S28, S31, S32, S33, S34] [41].

It is challenging to implement MPs in SMEs due to their limited resources [S13, S26, S27, S28, S31, S32, S33, S34] [41]. Most of project managers in SMEs perform measurement planning (e.g. estimating budget, schedule and effort) based on their experience and knowledge from previous projects [S4, S9, S18, S27, S32], [68].

Use of automated tools: In SLR on software MPs [6], the automated tools are divided into two main categories:

1. Tools that are specifically developed for measurement processes. These tools (e.g. StepCounter, Workflow, Eclipse Metrics plug-in) also help to provide data for effective measurement implementation.
2. Tools that are a part of the processes of any organization, e.g. project management, quality assurance. These tools are usually part of the whole management information system. The limitations of such tools include lack of metrics data exchange formats, effective usage of collected data to feed the decision making process, and using collected data to effectively monitor and control the software development processes.

In [S19], project management officers used Tarc (self-assessment tool) for the selection and collection of metrics based on the predefined data collection procedure. They defined 10 fundamental metrics and 7 derived metrics (e.g. productivity, effort per day, review density, problem density, test density, bug density) to measure size, quality and effort attributes using Tarc [S19]. The collected metrics were used for quality assurance.

The SCAPT tool [S24] measures the performance of SMEs based on time, cost and reliability of software production. SCAPT depends upon the availability of the company's own data collection procedure. It is tested on 44 different SMEs and it is observed that the unavailability of assessment data is a major hindrance for performance estimation.

The SonarQube tool is proposed to collect and analyse measurement data on software quality assurance practices in SMEs [S25]. Its objective is to continuously monitor a source code for problems such as code smells, antipatterns, and unused methods. The best practices of software quality assurance based on literature and experiences are maintained in the tool.

The SPIALS tool [S35] is based on the Standard CMMI Appraisal Method for Process Improvement (SCAMPI). Its objective is to assess SPI by using the lightweight CMMIbyScrum model. It measures SPI by conducting a survey with the help of a structured questionnaire that is based on the CMMIbyScrum model.

Comparison of measurement programs in SMEs and large organizations

In [6], the authors performed an SLR on software MPs and observed that 4 out of 65 primary studies focused on the MPs in SMEs. Therefore, we conducted this SMS to analyse factors, such as measurement models, challenges and metrics selection methods for implementing MPs in SMEs.

In this section, a comparison between software MPs in SMEs and large companies is presented. The SLR [6] identified 35 measurement models and 11 tools and SMS identified 29 measurement models and 4 tools. There are 4 measurement models in SLR that are proposed for SMEs, i.e. SPGQM [69], OMSD [9], MIS-PyME [8], and GQM-DSFMS [70] and these four models are identified as common between both studies. All of these four models are based on goal-oriented approaches.

The measurement models are categorized into "goal oriented approach improvement (GOAI)", "software process improvement (SPI)" and "measurement process improvement (MPI)" in both studies. Figure 9 shows that the majority of measurement models in the SLR are GOAI followed by MPI and SPI. On the other hand, the majority of measurement models in SMS are SPI followed by GOAI and MPI.

The metrics selection methods are categorized into "use of measurement standards", "use of measurement experts and experiences" and "use of automated tools" in both studies. The SLR [6] and SMS analysed a different number of primary studies; therefore, the frequencies and percentages of primary studies discussing these standards are presented in Figure 10a and Figure 10b.

One of the reasons for the disparity in the number of studies between SLR and SMS might be the late evolution of SMEs industry in the last two decades. The history of software measurement and how it became critical of SMEs is discussed at the beginning of Section 4. The primary studies in the SLR [6] discussed ISO/IEC 15939:2007 [55], ISO/IEC 25000 [56], ISO/IEC 9126-x [57], and ISO/IEC 14598-x [58], ISO/IEC/IEEE 24765:2010 [59], CMMI [60, 61], ISO/IEC 25021 [62], and ISO 9126 standard family [63–65]. On the other hand, in this SMS there were measurement models proposed to adapt guidelines and methods of those measurement models that are already reported for large organizations [6, 20]. The MIS-PyME [S2], SQIP [S6], PRISMS [S8], MESOPYME [S9], AAHA [S11], ASPISME [S27], and CMMIbyScrum [S35] models should adapt the CMMI standard in SMEs. Irrazabal et al. proposed guidelines to adapt ISO/IEC 12207:2008 standard for SCRUM [S28] and María et al. proposed guidelines to adapt ISO/IEC 15939:2007, ISO/IEC 12207:2008 and CMMI to SMEs [S29]. Pino et al. in an SLR [23] considered that ISO and SEI standards for SPI are not directly suitable for SMEs due to the complexity of recommendations and the requirement of a large investment of time and resources. Therefore, there is a need for widely accepted strategies to adapt these standards in SMEs. The organizations that develop international Software Engineering standards should separately consider implementing measurement processes in SMEs [23].

An MP was divided into three phases for further analysis. These phases are the pre-implemen-

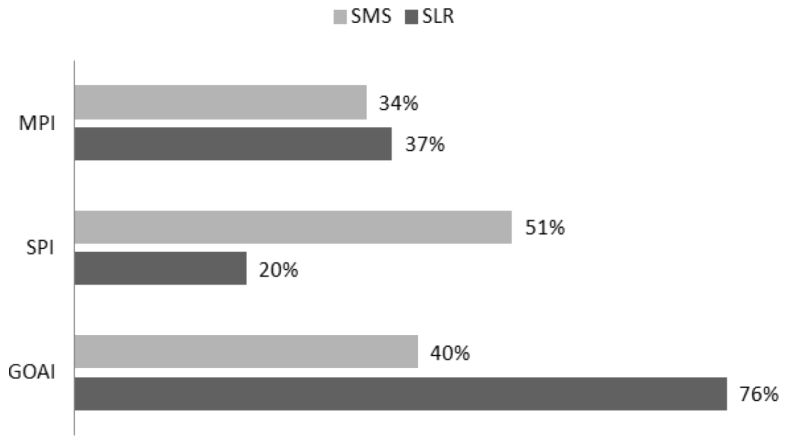
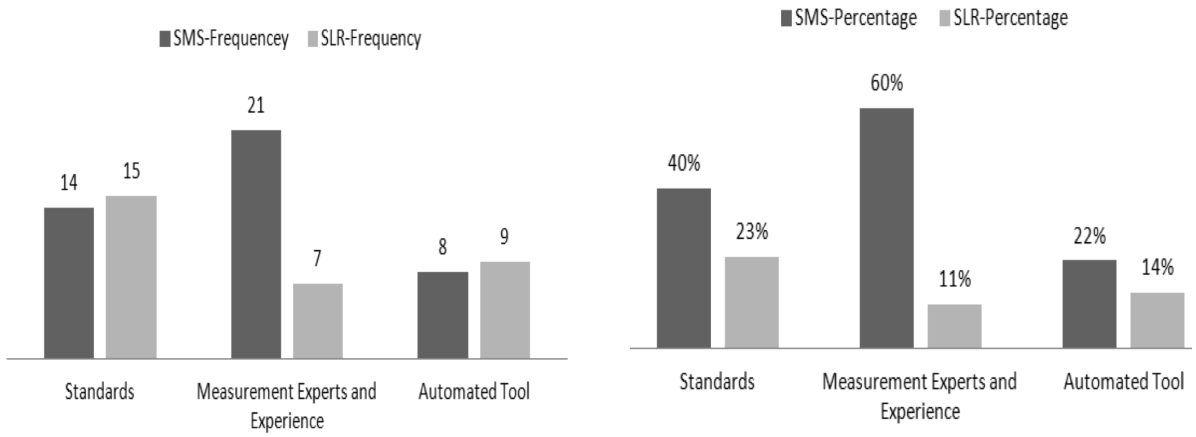


Figure 9. Comparison of categories of measurement models between SLR and SMS



(a) Metrics selection methods with respect to frequencies of primary studies discussing metric selection methods

(b) Metrics selection methods with respect to percentages of primary studies discussing metric selection methods

Figure 10. Metrics selection methods

tation, implementation and post-implementation of a MP.

Table 10 presents measurement purposes with respect to the phases of implementing an MP as shown in Figure 9. The pre-implementation phase of an MP starts with planning a software development process. In this phase, historical data from previous projects, measurement standards, measurement experts and experiences and automated tools might be used to predict the attributes of processes, products and resources (the details are in the results and analysis of RQ3). The implementation phase of an MP includes the characterization of issues/problems during software development life cycle and the continuous evaluation of project progress with respect to plans and predictions. The post-implementation phase of an MP helps in software process improvement based on lessons learned during the pre-implementation and implementation phase. The improvements can be twofold: 1) improvement in measurement processes, 2) improvement in software development processes. The prediction is the least utilized purpose among primary studies of SMS and SLR as shown in Table 10.

The measurement models for SMEs are specifically designed to implement the measurement process keeping the basic limitations of SMEs, such as budget, time, resources and low process maturity, in view. The measurement models proposed for large companies focus on broad issues, such as the measurement of customer satisfaction,

Table 10. Purposes of measurement program

Measurement studies	Pre-implementation prediction	Implementation characterization	Implementation evaluation	Post-implementation improvement
SLR	28%	81%	77%	70%
SMS	16%	63%	83%	93%

Table 11. Metrics discussed among primary studies of SMS and SLR

Measurement process	Measurement attributes	Metrics type	SMS		SLR	
			Frequency	Percentage	Frequency	Percentage
Pre-implementation	Size	Product	8	22.5%	10	15.4%
	Duration	Process	14	40%	7	10.7%
	Effort	Resource	13	37.1%	11	16.9%
	Cost/Budget	Process	-	-	4	6.1%
	Time to market	Product	-	-	3	4.6%
Implementation	Productivity	Resource	15	42.58%	11	16.9%
	Traceability	Product	5	14.2%	-	-
	Cyclomatic complexity	Product	4	11.4%	-	-
	Employee Commitment	Resource	-	-	8	12.3%
Post-implementation	Return on investment	Product	-	-	3	4.6%
	Customer satisfaction	Product	13	37.1%	8	12.3%
	Defects	Product	14	40%	25	38.5%
	Reliability	Product	6	17.1%	-	-

effectiveness of decisions taken based on MPs, verification and validation of the metrics collection process, building an information system for the measurement process, and the improvement of software development processes [6].

Table 11 presents the most commonly used metrics based on how frequently they are discussed in the primary studies in SMS and SLR. Fenton and Bieman [5] distinguished three types of measurement entities, i.e. process, product, and resource. Table 11 shows that the product metrics are mostly measures in the primary studies of SLR [6] and SMS. It also points out the need for more utilization of process and resource metrics for planning, organizing, monitoring, and controlling the processes and resources.

In SLR [6], they found that there is a lack of discussion of real-time metrics among primary studies (e.g. cyclomatic complexity, dynamic function calls, number of unused objects) to monitor and control the actual software development progress. Soini [71] conducted an empirical case study in the Finnish software industry to evaluate the actual use of software metrics. The software metrics are categorized into real-time and lagging metrics [71]. The real-time metrics help to monitor and control the ongoing processes in software organizations and provide indicators (e.g. cyclomatic complexity and traceability in this SMS). The lagging metrics are collected at the completion of projects (e.g. return on investment and customer satisfaction in this SMS). The balance between real time and lagging metrics might assist improvement in measurement processes [71].

Table 11 shows that all three types of metrics (i.e. process, product and resource) are only discussed for the pre-implementation phase of MPs. Furthermore, the process and product types of metrics are discussed twice as resource metrics in the pre-implementation phase. The resource and product types of metrics are discussed for the implementation phase of MPs and only product type of metrics are discussed in the post-implementation phase. The measurement of software defects is the most commonly discussed metric in both studies.

Table 12. Comparison of the challenges of implementing measurement programs in this SMS and SLR [6]

Challenges reported in SMS	Challenges reported in SLR
Pre-implementation – Lack of budget, time and resources allocated for software measurement. – Use of metrics is limited due to lack of experienced professionals. – Lack of measurement experts. – Lack of measurement maturity for implementing software MPs. – Absence of documentation and formal process management techniques. – Lack of automated tools for data collection. – Metrics are not validated for use in SMEs. Implementation – Scope of database containing indicators and measures is small as limited number of metrics are utilized in SMEs. – Limited utilization of metrics due to lack of defined process for management of quality issues in development process.	Pre-implementation – Lack of benchmarks. – Heterogeneity of SDLCs, MPs, products, culture, and priorities. Implementation – Correctness of MPs objectives. – Prioritisation of goals. – Transition to measurement culture. – Construct validity issues of metrics. – Lack of consistent definitions of measurement entities, tasks and processes. – Sync between MPs and SPI activities. – Overlapping between the metrics types. – Scalability issues in MPs. – Identification of correct measurement instrument. – Completeness, integrity, consistency of measurement data. – Lack of suitable metrics selection methods. – Lack of real time metrics (e.g. cyclomatic complexity, dynamic function calls, no of unused objects and variables) to monitor and control the actual software development progress. Post-implementation – Sustainability of MPs.

Table 12 presents the challenges of implementing MPs in SMEs and large organizations. The challenges are presented with respect to pre-implementation, implementation and post-implementation phases of an MP.

Pre-implementation challenges: The challenges which already exist in the software development organization (e.g. lack of budget, and time) or they exist in the software measurement domain (e.g. inconsistent measurement terminologies) before the implementation of MPs.

Implementation challenges: The challenges which appear during the implementation of MPs.

Post-implementation challenges: The challenges which appear after the implementation of MPs.

In the primary studies of SMS, most of the reported challenges exist even before the implementation of MPs in SMEs. They are of fundamental significance and encompass, e.g. lack of budget, time and resources. The SMEs usually hire fresh or less experienced graduates, which causes the lack of understanding and attention towards software quality and measurement issues [10,46]. The lack of defined measurement processes results in a situation when it is the higher management to decide on the importance of MPs and consequently the mechanism becomes people-oriented instead of process-oriented [S9, S18, S27]. The absence of formal documentation and automated measurement tools also hinders measurement processes because both are key sources to provide data for measurement [S19, S24]. It is also critical to learn whether the measured values are exactly the ones that were to be measured [72, 73]. The lack of metrics validation also imposes a challenge, as metrics must be mathematically correct and useful for decision-making [74], [S18, S19, S20, S25, S26, S27, S33].

The challenges faced during the implementation of MPs in SMEs include a limited scope of measurement repository (database) in terms of using metrics for the characterization, evaluation, prediction and improvement of software

entities at project and organization level [S5, S25, S29]. There are only few fundamental metrics which are used mostly by SMEs to plan, monitor and control software entities such as processes, products and resources [S22, S23].

The challenges reported by studies in large companies are mostly related to the issues discovered while implementing MPs [6]. The primary studies in SLR [6], report the lack of measurement benchmarks in terms of publically available measurement datasets, measurement standards and widely accepted measurement models and tools.

The heterogeneity of software organizations might be a challenge for implementing MPs in both SMEs and large organizations, e.g. in terms of software development life cycle (waterfall, agile etc.), size of organization (small, medium and large), domain of software products (e-commerce, mainframe systems, etc.), implementation levels of MPs (project or organization-wide), measurement purposes (characterize, evaluate, predict and/or improve) and measurement culture [6,20].

Construct validity is also a key challenge while implementing an MP, however, it was not possible to find specific discussions or solutions presented to address this challenge in SMEs. Kaner defined construct validity as, "How do you know that you are measuring what you think you are measuring" [73]. The software measurement is defined as the empirical, objective assignment of numbers according to a theory or model, to characterize the attribute of processes, products and resources [73]. In an SLR on the validation of software metrics [74], the word "construct" is referred to as a tool, instrument or procedure used to collect metrics. There are 47 validation criteria of software metrics presented in the SLR [74], however, they need further evaluations by researchers and practitioners to select suitable metrics validation criteria for measurement processes in large and SMEs industry. In this study 53 citations of the SLR [74] using Google Scholar were found, however, none of these specifically focused on metrics validation for SMEs.

Table 13 presents the comparisons of the implementation of MPs at project and organization level in the measurement studies of SLR [6] and SMS. According to both studies, it is challenging for software development organizations to implement MPs at both levels [6]. It might be due to the fact that most of the measurement models are designed to solve a specific problem at project level or organization level and their implementation is usually limited to a specific project. These factors might hinder the continuity of MPs for a longer period of time and at both implementation levels of MPs. Furthermore, 51% of the primary studies in SLR [6] and SMS are case studies. It is considered in [6,20,23] that there is a lack of comparative case studies of MPs. One of the potential reasons might be the fact that there is no clear context description in the published case studies. The context description might include organizational context of case studies, such as type and size of organization, type of products, measurement stakeholders. The description of the measurement process might include the type of metrics collected and the analysed, duration of measurement processes, analysis methodologies, link between measurement processes and improvement activities [6]. A comprehensive context description will help practitioners and researchers to achieve the repeatability, extensibility, and comparisons of case studies [6,20].

Table 13. Comparison of the implementation levels of measurement programs

Implementation levels of MPs	SLR	SMS
Project	58%	30%
Organization	28%	30%
Project AND Organization	14%	40%

The challenges faced during implementation of MPs at SMEs include limited scope of measurement repository (database) in terms of using metrics for characterization, evaluation, prediction and improvement of software entities at project and organization level [S5, S25, S29]. There are only few fundamental metrics which are used mostly by SMEs to plan, monitor and control software entities such as processes, products and resources [S22, S23]. These challenges exist even before the implementation of MPs at SMEs.

In SLR [6], the incremental development of MPs is also mentioned as a solution for software

organizations having no or partially defined MPs [52]. It was not possible to find any solution to the problem of initiating a measurement process in this mapping study. However, it was found in the SLR [6] that Iversen and Mattiassen [51] discussed the experiences of establishing an MP with the help of the incremental application of GQM and the intelligent collection and analysis of data. Therefore, the automation of the data collection process can be implemented incrementally. The first step may include the collection of data with manual entries into a measurement repository using a tool. In the second step, data collection may also be automated. This requires the integration of the MP with the SDLC [S4, S25, S28, S29, S30, S33, S34, S35]. There are both open source and commercial tools to automate the data collection for SDLC processes [52]. The use of automated tools for the characterization, evaluation, and prediction of software processes, products and resources becomes even more important in SMEs because there is a shortage of time, human and financial resources.

Large organizations mostly report challenges observed during the implementation of MPs while SMEs report pre-implementation challenges (e.g. budget, time, lack of measurement process maturity). The literature lacks challenges and mitigation strategies while implementing MPs at SMEs. Therefore, the SMEs can also evaluate mitigation strategies for the challenges presented in [6] according to their needs while implementing MPs.

5. Conclusion

The systematic mapping process proposed by Petersen et al. [21] is used to conduct this Systematic Mapping Study (SMS) [21]. The main objective of this mapping study is to identify and analyse the studies on software measurement programs (MPs) in small and medium enterprises (SMEs). In total, 35 primary studies are analysed to answer the following research questions:

RQ1: What measurement models, tools and practices for implementing measurement programs in SMEs are discussed in literature?

RQ2: What are the problems, challenges and issues of implementing measurement programs in SMEs?

RQ3: What metrics selection techniques, methods and approaches are used for measurement programs in SMEs?

This SMS analyses 29 measurement models and 4 tools. The measurement models are categorized into "goal oriented approach improvement (GOAI)", "software process improvement (SPI)" and "measurement process improvement (MPI)". The majority of the measurement models are built upon SPI (51%) approaches followed by GOAI (40%) and MPI (34%) approaches. As for the implementation level of MPs, most measurement models are implemented at both the project and organization level (40%) followed by project level (30%) and organization level (30%). With respect to the measurement purposes of models, the distribution of MPs is identified as: characterization (63%), evaluation (83%), improvement (93%) and prediction (16%). When the combination of purposes (i.e. when more than one purpose was mentioned by a primary study) was investigated, it was found out that around 59% of the studies mentioned the purposes of characterization and improvement while only 17% referred to all four purposes. This situation might be due to the fact that prediction based on historical data is possible if an MP lasts longer than a single project.

The metrics selection methods in primary studies are categorized into "use of measurement standards", "use of measurement experts and experiences" and "use of automated tools". The majority of primary studies discussed the use of measurement experts and experience (60%) followed by the use of measurement standards (40%) and the use of automated tools (22%). The common types of metrics discussed in the primary studies include productivity (43%), defects (40%), duration (40%), effort (37%), customer satisfaction (37%), size (22%), and cyclomatic complexity (11%). The most commonly used research methods in primary studies are a case study (51%) and a survey (25%). Most of the primary studies (80%) were published between 2006 and 2013.

Most of the SMEs face challenges, such as low measurement process maturity, limited resources to develop MPs and short time-to-market. Furthermore, the lack of measurement planning, tool support for data collection and measurement professionals are key challenges for the implementation of MPs.

In this study, the MPs in SMEs and large organizations are also compared. Most of the measurement models for SMEs are built upon the software process improvement approach. On the other hand, most of measurement models for large organizations are built upon goal-oriented approaches. The measurement models in SMS and SLR [6] focus the least on using measurement data for prediction. There is a lack of automated tools support for implementing MPs as there are 11 and 4 tools identified for large organizations and SMEs, respectively.

The SMEs and large organization face different challenges as studies in SMEs report challenges that existed even before the implementation of MPs due to different infrastructure and management processes of SMEs. Therefore, lightweight measurement models are proposed to cater for measurement processes while keeping the limitations of SMEs, such as budget, time and resources, in view. In this SMS, we found the measurement models which are proposed to adapt the guidelines and methods of those measurement models that are already reported for large organizations [6,20]. For instance, the MIS-PyME [S2], SQIP [S6], PRISMS [S8], MESOPYME [S9], AAHA [S11], ASPISME [S27], and CMMIbyScrum [S35] models are proposed to adapt the CMMI standard in SMEs. On the other hand, the challenges reported by studies in large companies are mostly related to the issues discovered while implementing MPs. These measurement studies report challenges, such as lack of measurement benchmarks in terms of measurement datasets, standards and widely accepted measurement models and tools. The challenges also include the lack of synchronization among measurement processes, software development processes and software improvement processes, and the adoption of measurement culture.

This SMS presented the findings from the existing literature. We are currently conducting online surveys in SMEs to validate the findings of SMS.

References

[1] P. Cocca and M. Alberti, "A framework to assess performance measurement systems in SMEs," *International Journal of Productivity and Performance Management*, Vol. 59, No. 2, 2010, pp. 186–200.

[2] U. Loecher, "Small and medium-sized enterprises – Delimitation and the European definition in the area of industrial business," *European Business Review*, Vol. 12, No. 5, 2000, pp. 261–264.

[3] M. Ayyagari, T. Beck, and A. Demirguc-Kunt, "Small and medium enterprises across the globe," *Small Business Economics*, Vol. 29, No. 4, 2007, pp. 415–434.

[4] M. Khalique, N. Bontis, J. Abdul Nassir bin Shaari, and A. Hassan Md. Isa, "Intellectual capital in small and medium enterprises in Pakistan," *Journal of Intellectual Capital*, Vol. 16, No. 1, 2015, pp. 224–238.

[5] N. Fenton and J. Bieman, *Software metrics: A rigorous and practical approach*. CRC Press, 2014.

[6] T. Tahir, G. Rasool, and C. Gencel, "A systematic literature review on software measurement programs," *Information and Software Technology*, Vol. 73, 2016, pp. 101–121.

[7] M. Díaz-Ley, F. García, and M. Piattini, "Implementing a software measurement program in small and medium enterprises: A suitable framework," *IET Software*, Vol. 2, No. 5, 2008, pp. 417–436.

[8] M. Díaz-Ley, F. García, and M. Piattini, "MIS-PyME software measurement capability maturity model – Supporting the definition of software measurement programs and capability determination," *Advances in Engineering Software*, Vol. 41, No. 10, 2010, pp. 1223–1237.

[9] A.M. Bhatti, H.M. Abdullah, and C. Gencel, "A model for selecting an optimum set of measures in software organizations," in *European Conference on Software Process Improvement*. Springer, 2009, pp. 44–56.

[10] M.K. Sharma, R. Bhagwat, and G.S. Dangayach, "Practice of performance measurement: experience from indian SMEs," *International*

Journal of Globalisation and Small Business, Vol. 1, No. 2, 2005, pp. 183–213.

[11] V. Claudia, M. Mirna, and M. Jezreel, "Characterization of software processes improvement needs in SMEs," in *International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE*. IEEE, 2013, pp. 223–228.

[12] I. Richardson and C.G. Von Wangenheim, "Guest editors' introduction: Why are small software organizations different?" *IEEE Software*, Vol. 24, No. 1, 2007, pp. 18–22.

[13] C.Y. Laporte, S. Alexandre, and R.V. O'Connor, "A software engineering lifecycle standard for very small enterprises," *Software Process Improvement*, 2008, pp. 129–141.

[14] B. Kitchenham, "What's up with software metrics? – a preliminary mapping study," *Journal of Systems and Software*, Vol. 83, No. 1, 2010, pp. 37–51.

[15] O. Gómez, H. Oktaba, M. Piattini, and F. García, "A systematic review measurement in software engineering: State-of-the-art in measures," in *International Conference on Software and Data Technologies*. Springer, 2006, pp. 165–176.

[16] C. Catal and B. Diri, "A systematic review of software fault prediction studies," *Expert Systems with Applications*, Vol. 36, No. 4, 2009, pp. 7346–7354.

[17] R. Malhotra, "A systematic review of machine learning techniques for software fault prediction," *Applied Soft Computing*, Vol. 27, 2015, pp. 504–518.

[18] D. Radjenović, M. Heričko, R. Torkar, and A. Živkovič, "Software fault prediction metrics: A systematic literature review," *Information and Software Technology*, Vol. 55, No. 8, 2013, pp. 1397–1418.

[19] T. Hall, S. Beecham, D. Bowes, D. Gray, and S. Counsell, "A systematic literature review on fault prediction performance in software engineering," *IEEE Transactions on Software Engineering*, Vol. 38, No. 6, 2012, pp. 1276–1304.

[20] M. Unterkalmsteiner, T. Gorschek, A.M. Islam, C.K. Cheng, R.B. Permadi, and R. Feldt, "Evaluation and measurement of software process improvementóa systematic literature review," *IEEE Transactions on Software Engineering*, Vol. 38, No. 2, 2012, pp. 398–424.

[21] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering." in *EASE*, Vol. 8, 2008, pp. 68–77.

[22] M. Sulayman and E. Mendes, "A systematic literature review of software process improvement in small and medium web companies," *Advances in Software Engineering*, 2009, pp. 1–8.

[23] F.J. Pino, F. García, and M. Piattini, "Software process improvement in small and medium software enterprises: A systematic review," *Software Quality Journal*, Vol. 16, No. 2, 2008, pp. 237–261.

[24] A. Ahmad and M.A. Babar, "Software architectures for robotic systems: A systematic mapping study," *Journal of Systems and Software*, Vol. 122, 2016, pp. 16–39.

[25] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, Vol. 64, 2015, pp. 1–18.

[26] F. García, M.F. Bertoa, C. Calero, A. Vallecillo, F. Ruíz, M. Piattini, and M. Genero, "Towards a consistent terminology for software measurement," *Information and Software Technology*, Vol. 48, No. 8, 2006, pp. 631–644.

[27] S. Keele *et al.*, "Guidelines for performing systematic literature reviews in software engineering," in *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*. sn, 2007.

[28] J.W. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage Publications, 2013.

[29] N. Mack, C. Woodsong, K.M. MacQueen, G. Guest, and E. Namey, "Qualitative research methods: A data collectors field guide." *POPLine*, 2005.

[30] R. Conradi and A.I. Wang, *Empirical methods and studies in software engineering: experiences from ESERNET*. Springer, 2003, Vol. 2765.

[31] N.E. Fenton and M. Neil, "Software metrics: Successes, failures and new directions," *Journal of Systems and Software*, Vol. 47, No. 2, 1999, pp. 149–157.

[32] G. Tom, *Software Metrics*. Chartwell-Bratt, 1976.

[33] R.B. Grady and D.L. Caswell, *Software metrics: Establishing a company-wide program*. Prentice Hall, 1987.

[34] V.R. Basili, G. Caldiera, and H.D. Rombach, "The goal question metric approach," in *Encyclopedia of Software Engineering*. Wiley, 1994, pp. 528–532.

[35] R.E. Park, W.B. Goethert, and W.A. Florac, "Goal-driven software measurement. A guidebook," Carnegie Mellon University, Pittsburgh, PA, USA, Tech. Rep., 1996.

[36] L.C. Briand, C.M. Differding, and H.D. Rombach, "Practical guidelines for measurement-based process improvement," *Software Process Improvement and Practice*, Vol. 2, No. 4, 1996, pp. 253–280.

[37] Ø. Moen, M. Gavlen, and I. Endresen, "Internationalization of small, computer software firms: Entry forms and market selection," *European Journal of Marketing*, Vol. 38, No. 9/10, 2004, pp. 1236–1251.

[38] C. Larman and V.R. Basili, "Iterative and incremental developments. A brief history," *Computer*, Vol. 36, No. 6, 2003, pp. 47–56.

[39] A. Cockburn, *Agile software development*. Addison-Wesley Boston, 2002, Vol. 177.

[40] A. Boyd, "The goals, questions, indicators, measures (GQIM) approach to the measurement of customer satisfaction with e-commerce Web sites," in *Aslib proceedings*, Vol. 54. MCB UP Ltd, 2002, pp. 177–187.

[41] J. Motwani, "Critical factors and performance measures of TQM," *The TQM magazine*, Vol. 13, No. 4, 2001, pp. 292–300.

[42] A. Abran, J.W. Moore, P. Bourque, and R. Dupuis, Eds., *Guide to the Software Engineering Body of Knowledge (SWEBOK–2004 Version)*. IEEE Computer Society, 2004.

[43] A. Abran, L. Buglione, and A. Sellami, "Software measurement body of knowledge – initial validation using Vincenti's classification of engineering knowledge types," in *Software Measurement Conference*, 2004, pp. 1–16.

[44] S. Alexandre, A. Renault, and N. Habra, "POWPL: A gradual approach for software process improvement in SMEs," in *32nd EUROMICRO Conference on Software Engineering and Advanced Applications*. IEEE, 2006, pp. 328–335.

[45] A.H. Lee, W.C. Chen, and C.J. Chang, "A fuzzy AHP and BSC approach for evaluating performance of IT department in the manufacturing industry in Taiwan," *Expert Systems with Applications*, Vol. 34, No. 1, 2008, pp. 96–107.

[46] P. Garengo, S. Biazzo, and U.S. Bititci, "Performance measurement systems in SMEs: A review for a research agenda," *International Journal of Management Reviews*, Vol. 7, No. 1, 2005, pp. 25–47.

[47] O.T. Pusatli, "Software measurement activities in small and medium enterprises: An empirical assessment," *Acta Polytechnica Hungarica*, Vol. 8, No. 5, 2011, pp. 21–42.

[48] C. Wang, E. Walker, and J. Redmond, "Explaining the lack of strategic planning in SMEs: The importance of owner motivation," *International Journal of Organisational Behaviour*, Vol. 12, No. 1, 2007, pp. 1–16.

[49] J. Chen, "Development of Chinese small and medium-sized enterprises," *Journal of Small Business and Enterprise Development*, Vol. 13, No. 2, 2006, pp. 140–147.

[50] M. Hudson, A. Smart, and M. Bourne, "Theory and practice in SME performance measurement systems," *International Journal of Operations & Production Management*, Vol. 21, No. 8, 2001, pp. 1096–1115.

[51] J. Iversen and L. Mathiassen, "Cultivation and engineering of a software metrics program," *Information Systems Journal*, Vol. 13, No. 1, 2003, pp. 3–19.

[52] B. Daubner, "Empowering software development environments by automatic software measurement," in *11th International Symposium Software Metrics*. IEEE, 2005, p. 3.

[53] A. Coletta, "An industrial experience in assessing the capability of non-software processes using ISO/IEC 15504," *Software Process: Improvement and Practice*, Vol. 12, No. 4, 2007, pp. 315–319.

[54] H. Abushama, M. Ramachandran, and P. Allen, *PRISMS: an approach to software process improvement for small to medium enterprises*. UOFK, 2016.

[55] *Systems and software engineering – Measurement process*, International Organization for Standardization Standard ISO/IEC 15 939:2007, 2007.

[56] *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Quality measure elements*, International Organization for Standardization Standard ISO/IEC 25 021:2012, 2012.

[57] *Product quality – Part 1: Quality model*, International Organization for Standardization Standard ISO/IEC 9126-1:2001, 2001.

[58] *Information Technology – Software Product Evaluation – Parts 1-6*, International Organization for Standardization Standard ISO/IEC 14 598, 2001.

[59] *Systems and software engineering – Vocabulary*, International Organization for Standardization Standard ISO/IEC/IEEE 24 765:2010, 2010.

[60] "Capability maturity model integration (CMMI) (continuous representation)," Carnegie Mellon University, Tech. Rep. ICMU/SEI-2002-TR-011, 2002. [Online]. https://resources.sei.cmu.edu/asset_files/TechnicalReport/2002_005_001_14039.pdf

[61] "Capability maturity model integration (CMMI) (staged representation)," Carnegie Mellon University, Tech. Rep. CMU/SEI-2002-TR-012, SEI, 2002. [Online]. https://www.sei.cmu.edu/reports/02tr029.pdf

[62] *Software engineering: Software product quality requirements and evaluation (square) quality measure elements*, International Organization for Standardization Standard ISO/IEC 2502-1, 2005.

[63] *Product quality – Part 2: Quality model*, International Organization for Standardization Standard ISO/IEC 9126-2:2001, 2001.

[64] *Product quality – Part 3: Quality model*, International Organization for Standardization Standard ISO/IEC 9126-3:2001, 2001.

[65] *Product quality – Part 4: Quality model*, International Organization for Standardization Standard ISO/IEC 9126-4:2001, 2001.

[66] F.G. Wilkie, D. McFall, and F. McCaffery, "An evaluation of CMMI process areas for small-to medium-sized software development organisations," *Software Process: Improvement and Practice*, Vol. 10, No. 2, 2005, pp. 189–201.

[67] J.A. Williams, "The impact of motivating factors on implementation of ISO 9001:2000 registration process," *Management Research News*, Vol. 27, No. 1/2, 2004, pp. 74–84.

[68] M. Jørgensen, "A review of studies on expert estimation of software development effort," *Journal of Systems and Software*, Vol. 70, No. 1, 2004, pp. 37–60.

[69] T. Tahir and C. Gencel, "A structured goal based measurement framework enabling traceability and prioritization," in *6th International Conference on Emerging Technologies (ICET)*. IEEE, 2010, pp. 282–286.

[70] C. Gencel, K. Petersen, A.A. Mughal, and M.I. Iqbal, "A decision support framework for metrics selection in goal-based measurement programs: GQM-DSFMS," *Journal of Systems and Software*, Vol. 86, No. 12, 2013, pp. 3091–3108.

[71] J. Soini, "A survey of metrics use in finnish software companies," in *International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 2011, pp. 49–57.

[72] P. Carbone, L. Buglione, L. Mari, and D. Petri, "A comparison between foundations of metrology and software measurement," *IEEE Transactions on Instrumentation and Measurement*, Vol. 57, No. 2, 2008, pp. 235–241.

[73] C. Kaner *et al.*, "Software engineering metrics: What do they measure and how do we know?" in *10th International Software Metrics Symposium, METRICS*. IEEE Computer Society, 2004.

[74] A. Meneely, B. Smith, and L. Williams, "Validating software metrics: A spectrum of philosophies," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, Vol. 21, No. 4, 2012, p. 24.

[75] M. Díaz-Ley, F. García, and M. Piattini, "Implementing software measurement programs in non mature small settings," *Software Process and Product Measurement*, 2008, pp. 154–167.

[76] M. Díaz-Ley, F. García, and M. Piattini, "Software measurement programs in SMEs–defining software indicators: A methodological framework," *Product-Focused Software Process Improvement*, 2007, pp. 247–261.

[77] M. Diaz-Ley, F. García, and M. Piattini, "MIS-PyME software measurement maturity model-supporting the definition of software measurement programs," *Product-Focused Software Process Improvement*, 2008, pp. 19–33.

[78] A. Tosun, A. Bener, and B. Turhan, "Implementation of a software quality improvement project in an SME: A before and after comparison," in *35th Euromicro Conference on Software Engineering and Advanced Applications*. IEEE, 2009, pp. 203–209.

[79] E. Amrina and S.M. Yusof, "A proposed manufacturing performance measures for small and medium-sized enterprises (SMEs)," in *Proceedings of the 10th Asia Pacific Industrial Engineering and Management Systems (APIEMS) Conference*, 2009, pp. 623–629.

[80] J.A.C.M. Villalón, G.C. Agustín, T.S.F. Gilabert, A.D.A. Seco, L.G. Sánchez, and M.P. Cota, "Experiences in the application of software process improvement in SMEs," *Software Quality Journal*, Vol. 10, No. 3, 2002, pp. 261–273.

[81] I. Richardson and K. Ryan, "Software process improvements in a very small company," *Software Quality Professional*, Vol. 3, No. 2, 2001, pp. 23–35.

[82] F. McCaffery, M. Pikkarainen, and I. Richardson, "AHAA–Agile, hybrid assessment method for automotive, safety critical SMEs," in *Proceedings of the 30th International Conference on Software Engineering*. ACM, 2008, pp. 551–560.

[83] F.T. Shah, S. Shamail, and N. Ahmad Akhtar, "Lean quality improvement model for quality practices in software industry in Pakistan," *Journal of Software: Evolution and Process*, Vol. 27, No. 4, 2015, pp. 237–254.

[84] S. Bibi, I. Stamelos, G. Gerolimos, and V. Kollias, "BBN based approach for improving the software development process of an SME – A case study," *Journal of Software: Evolution and Process*, Vol. 22, No. 2, 2010.

[85] P. Clarke and R.V. OíConnor, "The meaning of success for software SMEs: An holistic scorecard based approach," in *European Conference on Software Process Improvement*. Springer, 2011, pp. 72–83.

[86] E. Caballero, J.A. Calvo-Manzano, and T. San Feliu, "Introducing scrum in a very small enterprise: A productivity and quality analysis," *Systems, Software and Service Process Improvement*, 2011, pp. 215–224.

[87] C.G. von Wangenheim, T. Punter, and A. Anacleto, "Software measurement for small and medium enterprises," in *Proceeding 7th International Conference on Empirical Assessment in Software Engineering (EASE)*, 2003.

[88] M. Sulayman, C. Urquhart, E. Mendes, and S. Seidel, "Software process improvement success factors for small and medium Web companies: A qualitative study," *Information and Software Technology*, Vol. 54, No. 5, 2012, pp. 479–500.

[89] N. Ohsugi, K. Fushida, N. Inoguchi, H. Arai, H. Yamanaka, T. Niwa, M. Fujinuki, M. Tomura, and T. Kitani, "Using trac for empirical data collection and analysis in developing small and medium-sized enterprise systems," in *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 2015, pp. 1–9.

[90] H.M. Haddad and D.E. Meredith, "Instituting software metrics in small organizations: A practical approach," in *Eighth International Conference on Information Technology: New Generations (ITNG)*. IEEE, 2011, pp. 227–232.

[91] A. Potter, P. Childerhouse, R. Banomyong, and N. Supatn, "Developing a supply chain performance tool for SMEs in Thailand," *Supply Chain Management: An International Journal*, Vol. 16, No. 1, 2011, pp. 20–31.

[92] A. Janes, V. Lenarduzzi, and A.C. Stan, "A continuous software quality monitoring approach for small and medium enterprises," in *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion*. ACM, 2017, pp. 97–100.

[93] H. Ayed, N. Habra, and B. Vanderose, "AM-QuICk: A measurement-based framework for Agile methods customisation," in *Software Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA)*. IEEE, 2013, pp. 71–80.

[94] S. Suwanya and W. Kurutach, "Applying agility framework in small and medium enterprises," *Advances in Software Engineering*, 2009, pp. 102–110.

[95] E. Irrazabal, F. Vásquez, R. Díaz, and J. Garzás, "Applying ISO/IEC 12207:2008 with SCRUM and Agile methods," *Software Process Improvement and Capability Determination*, 2011, pp. 169–180.

[96] M. Díaz, F. García, and M. Piattini, "Defining, performing and maintaining software measurement programs: State of the art," in *IV Simposio Internacional de Sistemas de Informacióne*, 2006, p. 13.

[97] J.C. Ruiz, Z.B. Osorio, J. Mejia, M. Muñoz, A.M. Ch, B.A. Olivares *et al.*, "Definition of a hybrid measurement process for the models ISO/IEC 15504 – ISO/IEC 12207:2008 and CMMI Dev 1.3 in SMEs," in *Electronics, Robotics and Automotive Mechanics Conference (CERMA)*. IEEE, 2011, pp. 421–426.

[98] M. Jezreel, M. Mirna, N. Pablo, O. Edgar, G. Alejandro, and M. Sandra, "Identifying findings for software process improvement in SMEs: An experience," in *Ninth Electronics, Robotics and Automotive Mechanics Conference (CERMA)*. IEEE, 2012, pp. 141–146.

[99] F.J. Pino, F. Garcia, and M. Piattini, "Key processes to start software process improvement in small companies," in *Proceedings of the 2009 ACM symposium on Applied Computing*. ACM, 2009, pp. 509–516.

[100] R. Chalmeta, S. Palomero, and M. Matilla, "Methodology to develop a performance measurement system in small and medium-sized enterprises," *International Journal of Computer Integrated Manufacturing*, Vol. 25, No. 8, 2012, pp. 716–740.

[101] M. Lepmets and T. McBride, "Process improvement for the small and agile," in *European Conference on Software Process Improvement*. Springer, 2012, pp. 310–318.

[102] D. Homchuenchom, C. Piyabunditkul, H. Lichter, and T. Anwar, "SPIALS: A light-weight software process improvement self-assessment tool," in *5th Malaysian Conference in Software Engineering (MySEC)*. IEEE, 2011, pp. 195–199.

Appendix. List of selected studies

Paper ID	Title	Empirical method	Year
S1 [7]	M. Díaz-Ley, F. García, and M. Piattini, "Implementing a software measurement program in small and medium enterprises: A suitable framework," *IET Software*, Vol. 2, No. 5, 2008, pp. 417–436.	Case study	2008
S2 [8]	M. Díaz-Ley, F. García, and M. Piattini, "MIS-PyME software measurement capability maturity model–supporting the definition of software measurement programs and capability determination," *Advances in Engineering Software*, Vol. 41, No. 10, 2010, pp. 1223–1237.	Case study	2010
S3 [75]	M. Díaz-Ley, F. García, and M. Piattini, "Implementing software measurement programs in non mature small settings," *Software Process and Product Measurement*, 2008, pp. 154–167.	Industry report	2008
S4 [76]	M. Díaz-Ley, F. García, and M. Piattini, "Software measurement programs in SMEs–defining software indicators: A methodological framework," *Product-Focused Software Process Improvement*, 2007, pp. 247–261.	Industry report	2007
S5 [77]	M. Diaz-Ley, F. García, and M. Piattini, "MIS-PyME software measurement maturity model-supporting the definition of software measurement programs," *Product-Focused Software Process Improvement*, 2008, pp. 19–33.	Case study	2008
S6 [78]	A. Tosun, A. Bener, and B. Turhan, "Implementation of a software quality improvement project in an SME: A before and after comparison," in *35th Euromicro Conference on Software Engineering and Advanced Applications*. IEEE, 2009, pp. 203–209.	Industry report	2009
S7 [79]	E. Amrina and S.M. Yusof, "A proposed manufacturing performance measures for small and medium-sized enterprises (SMEs)," in *Proceedings of the 10th Asia Pacific Industrial Engineering and Management Systems (APIEMS) Conference*, 2009, pp. 623–629.	Survey	2009
S8 [54]	H. Abushama, M. Ramachandran, and P. Allen, *PRISMS: an approach to software process improvement for small to medium enterprises*. UOFK, 2016.	Survey	2016
S9 [80]	J.A.C.M. Villalón, G.C. Agustín, T.S.F. Gilabert, A.D.A. Seco, L.G. Sánchez, and M.P. Cota, "Experiences in the application of software process improvement in SMEs," *Software Quality Journal*, Vol. 10, No. 3, 2002, pp. 261–273.	Experiment	2002
S10 [81]	I. Richardson and K. Ryan, "Software process improvements in a very small company," *Software Quality Professional*, Vol. 3, No. 2, 2001, pp. 23–35.	Survey	2001
S11 [82]	F. McCaffery, M. Pikkarainen, and I. Richardson, "AHAA–Agile, hybrid assessment method for automotive, safety critical SMEs," in *Proceedings of the 30th International Conference on Software Engineering*. ACM, 2008, pp. 551–560.	Industry report	2008
S12 [83]	F.T. Shah, S. Shamail, and N. Ahmad Akhtar, "Lean quality improvement model for quality practices in software industry in Pakistan," *Journal of Software: Evolution and Process*, Vol. 27, No. 4, 2015, pp. 237–254.	Survey	2015
S13 [84]	S. Bibi, I. Stamelos, G. Gerolimos, and V. Kollias, "BBN based approach for improving the software development process of an SME – A case study," *Journal of Software: Evolution and Process*, Vol. 22, No. 2, 2010.	Case study	2010

Paper ID	Title	Empirical method	Year
S14 [85]	P. Clarke and R.V. OíConnor, "The meaning of success for software SMEs: An holistic scorecard based approach," in *European Conference on Software Process Improvement*. Springer, 2011, pp. 72–83.	Survey	2011
S15 [86]	E. Caballero, J.A. Calvo-Manzano, and T. San Feliu, "Introducing scrum in a very small enterprise: A productivity and quality analysis," *Systems, Software and Service Process Improvement*, 2011, pp. 215–224.	Experiment	2011
S16 [87]	C.G. von Wangenheim, T. Punter, and A. Anacleto, "Software measurement for small and medium enterprises," in *Proceeding 7th International Conference on Empirical Assessment in Software Engineering (EASE)*, 2003.	Experiment	2003
S17 [47]	O.T. Pusatli, "Software measurement activities in small and medium enterprises: An empirical assessment," *Acta Polytechnica Hungarica*, Vol. 8, No. 5, 2011, pp. 21–42.	Survey	2011
S18 [88]	M. Sulayman, C. Urquhart, E. Mendes, and S. Seidel, "Software process improvement success factors for small and medium Web companies: A qualitative study," *Information and Software Technology*, Vol. 54, No. 5, 2012, pp. 479–500.	Case study	2012
S19 [89]	N. Ohsugi, K. Fushida, N. Inoguchi, H. Arai, H. Yamanaka, T. Niwa, M. Fujinuki, M. Tomura, and T. Kitani, "Using trac for empirical data collection and analysis in developing small and medium-sized enterprise systems," in *Empirical Software Engineering and Measurement (ESEM), 2015 ACM/IEEE International Symposium on*. IEEE, 2015, pp. 1–9.	Case study	2015
S20 [70]	C. Gencel, K. Petersen, A.A. Mughal, and M.I. Iqbal, "A decision support framework for metrics selection in goal-based measurement programs: GQM-DSFMS," *Journal of Systems and Software*, Vol. 86, No. 12, 2013, pp. 3091–3108.	Case study	2013
S21 [90]	H.M. Haddad and D.E. Meredith, "Instituting software metrics in small organizations: A practical approach," in *Information Technology: New Generations (ITNG), 2011 Eighth International Conference on*. IEEE, 2011, pp. 227–232.	Industry report	2011
S22 [9]	A.M. Bhatti, H.M. Abdullah, and C. Gencel, "A model for selecting an optimum set of measures in software organizations," in *European Conference on Software Process Improvement*. Springer, 2009, pp. 44–56.	Survey	2009
S23 [69]	T. Tahir and C. Gencel, "A structured goal based measurement framework enabling traceability and prioritization," in *6th International Conference on Emerging Technologies (ICET)*. IEEE, 2010, pp. 282–286.	Case study	2010
S24 [91]	A. Potter, P. Childerhouse, R. Banomyong, and N. Supatn, "Developing a supply chain performance tool for smes in thailand," *Supply Chain Management: An International Journal*, Vol. 16, No. 1, 2011, pp. 20–31.	Survey	2011
S25 [92]	A. Janes, V. Lenarduzzi, and A.C. Stan, "A continuous software quality monitoring approach for small and medium enterprises," in *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion*. ACM, 2017, pp. 97–100.	Case study	2017

Paper ID	Title	Empirical method	Year
S26 [93]	H. Ayed, N. Habra, and B. Vanderose, "AM-QuICk: A measurement-based framework for Agile methods customisation," in *Software Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2013*. IEEE, 2013, pp. 71–80.	Case study	2013
S27 [94]	S. Suwanya and W. Kurutach, "Applying agility framework in small and medium enterprises," *Advances in Software Engineering*, 2009, pp. 102–110.	Case study	2009
S28 [95]	E. Irrazabal, F. Vásquez, R. Díaz, and J. Garzás, "Applying ISO/IEC 12207:2008 with SCRUM and Agile methods," *Software Process Improvement and Capability Determination*, 2011, pp. 169–180.	Case study	2011
S29 [96]	M. Díaz, F. García, and M. Piattini, "Defining, performing and maintaining software measurement programs: State of the art," in *IV Simposio Internacional de Sistemas de Informacióne*, 2006, p. 13.	Survey	2006
S30 [97]	J.C. Ruiz, Z.B. Osorio, J. Mejia, M. Muñoz, A.M. Ch, B.A. Olivares *et al.*, "Definition of a hybrid measurement process for the models ISO/IEC 15504 – ISO/IEC 12207:2008 and CMMI Dev 1.3 in SMEs," in *Electronics, Robotics and Automotive Mechanics Conference (CERMA)*. IEEE, 2011, pp. 421–426.	Case study	2011
S31 [98]	M. Jezreel, M. Mirna, N. Pablo, O. Edgar, G. Alejandro, and M. Sandra, "Identifying findings for software process improvement in SMEs: An experience," in *Ninth Electronics, Robotics and Automotive Mechanics Conference (CERMA)*. IEEE, 2012, pp. 141–146.	Case study	2012
S32 [99]	F.J. Pino, F. Garcia, and M. Piattini, "Key processes to start software process improvement in small companies," in *Proceedings of the 2009 ACM symposium on Applied Computing*. ACM, 2009, pp. 509–516.	Case study	2009
S33 [100]	R. Chalmeta, S. Palomero, and M. Matilla, "Methodology to develop a performance measurement system in small and medium-sized enterprises," *International Journal of Computer Integrated Manufacturing*, Vol. 25, No. 8, 2012, pp. 716–740.	Case study	2012
S34 [101]	M. Lepmets and T. McBride, "Process improvement for the small and agile," in *European Conference on Software Process Improvement*. Springer, 2012, pp. 310–318.	Case study	2012
S35 [102]	D. Homchuenchom, C. Piyabunditkul, H. Lichter, and T. Anwar, "SPIALS: A light-weight software process improvement self-assessment tool," in *5th Malaysian Conference in Software Engineering (MySEC)*. IEEE, 2011, pp. 195–199.	Case study	2012

e-Informatica Software Engineering Journal, Volume 12, Issue 1, 2018, pages: 167–198, DOI 10.5277/e-Inf180107

The Role of Organisational Phenomena in Software Cost Estimation: A Case Study of Supporting and Hindering Factors

Jurka Rahikkala*, Sami Hyrynsalmi**, Ville Leppänen***, Ivan Porres****

* *Vaadin Ltd*

** *Pervasive Computing, Tampere University of Technology*

*** *Department of Future Technologies, University of Turku*

**** *Department of Information Technologies, Åbo Akademi University*

jurka.rahikkala@vaadin.com, sami.hyrynsalmi@tut.fi, ville.leppanen@utu.fi, ivan.porres@abo.fi

Abstract

Context: Despite the fact that many researchers and practitioners agree that organisational issues are equally important as technical issues from the software cost estimation (SCE) success point of view, most of the research focus has been put on the development of methods, whereas organisational factors have received surprisingly little academic scrutiny.

Objective: This study aims to identify organisational factors that either support or hinder meaningful SCE, identifying their impact on estimation success. Top management's role is specifically addressed.

Methods: The study takes a qualitative and explorative case study based approach. In total, 18 semi-structured interviews aided the study of three projects in three organisations. Hence, the transferability of the results is limited.

Results: The results suggest that the role of the top management is important in creating prerequisites for meaningful estimation, but their day-to-day participation is not required for successful estimation. Top management may also induce undesired distortion in estimation. Estimation maturity and estimation success seem to have an interrelationship with software process maturity, but there seem to be no significant individual organisational factors, which alone would make estimation successful.

Conclusion: Our results validate several distortions and biases reported in the previous studies, and show the SCE research focus has remained on methodologies and technical issues.

Keywords: software cost estimation, project management, project success, top management, organisational factors, software improvement, software process maturity, case study

1. Introduction

Most software projects still suffer from budget and schedule overruns [1–4]. Regardless of the high price of software projects that bring hundreds of billions of euros in losses annually [5–7], there are still severe deficiencies in the proper application of software cost estimation methodologies in organisations [8–13].

Systematic overruns have continued for decades, although researchers and practitioners have developed hundreds of estimation methodologies [13, 14]. However, the reason for the overruns may not reside only in the estimation methodologies as they are shown to be able to produce accurate results when used properly [15, 16]. Thus, the problems that result in estimation errors may occur because estimation

methodologies are used ineffectively by organisations [9, 11, 14]. Consequently, organisational inhibitors [10], top management focus [17] and the sources of distortions [9,12] have become the focus of recent studies.

While most SCE does not use a proper methodology, the situation is considerably better in the area of project management (PM) as, according to Fortune and White [18], only 5% of projects do not use any PM tools. Considering the fact that cost estimation is an inseparable part of all projects [19], and that the cause of overruns in software projects may reside in software cost estimation (SCE), project management (PM) or other areas [20–22], the difference in the extent of the use of methodologies between software project management and management of other types of projects is surprising. Especially, because commonly used industrial project management and process improvement frameworks, such as CMMI [23], PMBOK [19] and IPMA ICB [24], promote the importance of estimation and the use of methodologies. The use of proper methodologies is proven to have a positive effect on the outcome of both SCE and PM [18, 25, 26], nevertheless only PM professionals utilise these valuable tools and methods to any great extent.

As scientific literature or industrial advice does not provide a clear explanation for the gap in the extent of the use of methodologies between SCE and PM, one assumption is that the difference arises from organisational priorities and does not seem to be related to the availability of proven cost estimation methodologies. Project management is widely linked to the execution of the corporate strategy [27–29], but SCE seems to have very little visibility among top management. Also, while project management research paid close attention to non-technical factors, such as top management support, communication, skills and learning [18, 30], SCE research mostly focused on developing and improving estimation techniques [14]. This is an important observation, indicating that the explanation for the difference in the extent of use of SCE and PM methodologies could reside within the research areas omitted from the study of SCE.

The goal of this study is to identify organisational factors that either support or hinder meaningful SCE, and to establish their impact on estimation success. The study takes a holistic view with special attention on top management participation. A qualitative, exploratory case study approach is employed, using interviews as the primary data collection method. In total, three projects were studied and 18 semi-structured interviews were conducted.

Some research papers addressing SCE from the organisational rather than technical viewpoint have been published recently [9, 10, 17, 31]. This paper continues on this highly relevant path but diverges from previous studies by studying the impact of organisational factors related to software process or project process on the effectiveness of the use of estimation methodologies. Improving the understanding of the real-world dynamics related to the effective use of estimation methodologies may provide practitioners with valuable tools for improving SCE in organisations. Especially, the gap between the advice provided by the industrial project management frameworks and the low extent of use of methodologies could be narrowed. This study may also provide further evidence that organisational issues are equally important as technical ones for effective SCE, and generate new theories about the reasons for why the extent of use of methodologies is low regardless of the experienced importance of SCE and industrial advise. This would justify further study on the organisational dimension of SCE.

The remaining part of the paper is structured as follows: Section 2 presents related work focusing on four areas: software cost estimation, project management, top management involvement and software cost estimation in industrial frameworks. Section 3 presents the research questions. Section 4 introduces the case companies and projects, and Section 5 elaborates on the research design. Section 6 presents the results of the case study and is followed by a discussion of the key findings in Section 7. Section 8 concludes the study.

Table 1. Distribution of research topics in software cost estimation. A single study can belong to multiple categories. Adapted from [14]

Perspective	–1989	1990–1999	2000–2004	Total
Estimation method	73%	59%	58%	61%
Size measures	12%	24%	16%	20%
Organisational issues	22%	15%	14%	16%
Uncertainty assessment	5%	6%	13%	8%
Calibration of models	7%	8%	4%	7%
Production function	20%	4%	3%	6%
Measures of estimation performance	5%	5%	6%	5%
Data set properties	0%	1%	2%	1%
Other	0%	2%	1%	1%

2. Related work

In the following subsections, top management's relationship to SCE and PM is reviewed and the focus areas of earlier research on these subjects is summarised.

2.1. Software cost estimation

Software cost estimation is an activity that aims to produce a prediction of the effort required to build a software component. As most costs in software development projects are personnel costs, 'cost' and 'effort' are often used interchangeably. The literature that studies and develops methods to estimate costs in software projects began in the 1960s [32, 33]. However, despite five decades of research and hundreds of studies [14, 34], software projects still exceed their budgets and timetables.

Jørgensen and Shepperd [14] conducted the most recent systematic literature review of SCE. In total, they selected 304 journal articles for their study and identified eight active research topics in SCE:

Estimation methods: the key issues include formal estimation models, expert estimation processes, decomposition based estimation processes and combinations of those three.

Production function: the key issues are the linear versus nonlinear relationship between effort and size, and the relationship between effort and schedule compression.

Calibration of models: the key issue is the calibration of estimation models, e.g. studies on local versus multi-organisational data and the calibration of the COCOMO model for certain types of projects.

Size measures: the key issues include validity and improvements in the size measures that are important in estimation models, e.g. the inter-rater validity of function point counting.

Organisational issues: the key issues are estimation processes in a wide organisational context, e.g. estimation practice, the reasons for cost overruns, the impact of estimates on project work, and estimation in the general context of project management.

Effort uncertainty assessment: the key issue is the uncertainty of effort or size estimates, e.g. methods providing minimum-maximum intervals for effort.

Measures of estimation performance: the key issues include the evaluation and selection of estimation methods, e.g. how to measure estimation accuracy or how to compare estimation methods.

Data set properties: the key issue is how to analyse data sets for the purpose of estimation methods, e.g. data sets with missing data.

Other: unclassified topics.

The distribution of the topics is presented in Table 1.

As shown in Table 1, all other categories except 'Organisational issues' and 'Other' focus on estimation methodologies or other formal methods for improving the estimation of size, effort or schedule. Only 16% of the articles discussed issues other than non-technical issues, i.e. organisational issues. Thus, SCE research strongly focuses on formal and technical issues and has

relatively little focus on non-technical topics. Furthermore, the share of the articles focusing on organisational issues seems to be decreasing, it was only 14% during the period from 2000 to 2004. The recent study of SCE research trends shows also that the research focus has remained consistently on estimation methodologies and techniques between 1996 and 2016, the emerged research areas being 'size metrics', 'estimation by analogy', 'tools for estimation', 'soft computing techniques' and 'expert judgement' in five topic solution [35].

Estimation methodologies produce good results when applied properly [15, 16]. Regardless of this, overruns still continue. While an obvious research topic should be the effective application of estimation methodologies, 84% of the articles still focus on improving methodologies. Hihn and Habib-agahi noticed already in 1991 that only 17% of the estimators used proper estimation methodologies [36]. This, however, seems not to have affected the research focus either. Also according to our experiences, the basic problem of SCE is that the estimation methodologies are not applied properly; researchers and practitioners largely agree on this point [13, 14]. Furthermore, Jørgensen and Shepperd's [14] review reports that only eight articles out of 304 were in-depth case studies and only three evaluated the background to the estimation processes. This, together with the technical focus of the research, confirms that concentrating on real-world issues that prevent the effective use of SCE methods is justified as a systematic improvement in SCE success that can only be realised through the successful application of estimation methods in real-world situations.

2.2. Project management

The share of work organised as projects is very high in organisations, and the results of such projects are critical for the success of an organisation [37,38]. Due to the significance of PM, the topic has been broadly studied and the body of knowledge on it is extensive. Several different categorisations of PM research areas exist and the following six perspectives have been presented by Kolltveit, Karlsen and Grønhaug [30]:

The task perspective: key issues include the scope of project management for a task, project targets, project results and planning and control.

The leadership perspective: key issues are leadership, communication, uncertainty and learning

The system perspective: key issues are systems, elements of systems, boundaries and dynamics.

The stakeholder perspective: key issues include stakeholders, communication, negotiation, relationships, influence and dependence.

The transaction cost perspective: key issues are transactions, transaction costs, production costs, and governance structure.

The business by project perspective: key issues include business, project results, project success, strategy, profit and benefits.

In their article, Kolltveit et al. [30] identified 562 articles published in *International Journal of Project Management* and classified them into the six above mentioned categories (see Table 2).

Once again, when dividing the areas or aspects into technical and non-technical, the task and transaction cost perspectives can be seen as technical. The other four can be seen as non-technical, or at least having most of their key issues beyond the purely technical focus. As Table 2 shows, the focus of the project management research was shifting from the task perspective towards the leadership and business perspectives. This can be seen from the table as with the above classification into technical and non-technical aspects, the share of technical perspectives decreased from 68% to 18% between the first and the last period, respectively. This shift of focus seems reasonable since organisational issues are reported to be even more important factors in project success than technical ones [25, 39–41]. Top management support (TMS) was even suggested as the most important factor affecting project success [42], which corresponds well with the largest share of the leadership perspective related papers.

Table 2. The distribution of research perspectives in project management. A single study can belong to multiple categories. Adapted from [30]

Perspective	1983–1987	1988–1992	1993–1997	1998–2002	2003–2004	Total
Task	49%	34%	32%	23%	12%	29%
Leadership	8%	16%	25%	28%	33%	23%
System	23%	25%	18%	19%	15%	20%
Stakeholder	1%	3%	1%	5%	6%	3%
Transaction	19%	9%	6%	10%	6%	10%
Business	0%	13%	17%	15%	29%	15%

In comparison to SCE research, PM research underwent a major shift from task oriented or technical topics towards people oriented or non-technical ones, whereas the SCE research focus remains on task oriented subjects. Thus, it is also reasonable to assume that the focus of PM research has placed more focus on how methods are applied by people and therefore increased the awareness, effectiveness and extent of use of the methods. The mere existence of a method seldom leads to its success.

2.3. Top management focus

Top management support has been found to be one of the most important critical success factors for project success in several studies [40, 42, 43] and few would doubt the need for TMS [44]. Also, top management's interest in PM is increasing along with the number of PM related articles published in top management and business journals [45]. However, top managers are generally more interested in non-technical issues of a strategic nature [46, 47].

The practices through which TMS is demonstrated for a project have been extensively studied. Garrity [48] recommends top management review plans and monitor results. Beath [49] found that top managers are able to make organisational changes, while Morton [50] notes top managers – as project champions – have the skills to mobilise public opinion, resolve conflicts between stakeholders and win the hearts and minds of project teams. Zwikael [25] identified a list of 10 critical top management support processes that influence a project's success, including appropriate PM assignment, project manager involvement during the initiation stage and the use of standard PM software.

TMS was not studied widely in the scope of SCE. However, Rahikkala et al. [17] found that top management pays attention to SCE and recognises that good estimates are critical for an organisation's success, as well as for understanding the consequences of an erroneous estimate. In general, there is very little information about TMS in SCE. This suggests that the actual top management focus on SCE is low. Regardless of the reported attention, the limited use of SCE methodologies supports this assumption.

2.4. Software cost estimation in industrial frameworks

Many of the commonly used project management frameworks, standards and other related guidelines address cost estimation. Project Management Institute's PMBOK Guide [19], as well as its Software Extension [51], give detailed guidance on preparing software cost estimates. Another popular framework, International Project Management Association's Competence Baseline [24], includes cost estimation as an important step. Furthermore, PRINCE2[1] and ITIL v3 [52] frameworks emphasize estimation and cost management, as well as the CMMI process improvement program [23] and the ISO 21500:2012 standard for project management [53]. Even the U.S. Government Accountability Office (GAO) published a 12 step guide for cost estimation[2].

[1] https://www.axelos.com/qualifications/prince2-qualifications
[2] http://energy.gov/sites/prod/files/GAO%2012-Step%20Estimating%20Process.pdf

Finally, cost estimation is also covered by agile methodologies [54].

3. Research questions

The literature review shows that SCE research has been centred on methodology for decades without a significant change. In contrast, PM research is very broad and covers topics such as methodologies, leadership and business. The focus of research also shifted from methodologies towards other areas, currently having a relatively even distribution on a broad range of topics. In particular, TMS was studied in the scope of PM but not SCE. Hence, though SCE and PM belong to software project delivery, the research focus is different. In the industrial context, the importance of SCE is widely recognized, and practically all major industrial bodies of knowledge provide guidance for cost estimation.

The above, together with the argument that proper cost estimation is often omitted [10, 36], suggests that the accountability of the use of meaningful estimation methodologies is unclear in organisations. There are no reports that SCE would be commonly omitted completely, rather that it is not conducted in a meaningful way. The previously reviewed project management and process improvement frameworks define clearly that project management is responsible for that the estimation is done, but not specifically that they would be responsible for how it is done. This seems to leave a gap in the software process, which may be one reason for malpractices and overruns. This motivates our first initial objective:

RQ1: What are the real-world factors concerning the organisational context of SCE (organisational factors) that either support or hinder the creation of a meaningful software cost estimate?

In our study, the organisational context refers widely to the properties and mechanisms of an organisation, such as top management commitment, leadership, organisational structure, communication, monitoring, recognition and education [55]. Effectively, the definition of the organisational context used in this study does not exclude any properties or mechanisms of an organisation, and we seek to identify the aspects affecting SCE that human subjects can or are willing to tell us about the topic [56]. Additionally, although the organisational context is the primary focus, biases emerging from human behaviour, as human subjects are centric for the organisational context, are also considered.

It has been found that technical issues are of little interest to senior managers [46, 47]. One reason for the existence of the previously described gap may be that SCE is perhaps perceived as too technical and too specific to software development to interest project managers. On the other hand, although software developers traditionally focus on technical topics and have little interest in or power over non-technical issues, they may not perceive SCE as a technical issue, and consider it as belonging to the project management's domain. Technical experts may also be protective of their domain in order to prevent loss of power to outsiders [57], while the suspicious and negative attitudes of senior managers towards IT and technical personnel [58] may hinder cooperation further. Therefore, the second initial objective of this study is to answer the second research question:

RQ2: What is the impact of top management in either supporting or hindering software cost estimation practices?

Finally, this paper draws attention to the difference between the extent of the use of SCE and PM methodologies, as well as to the different focus areas of research on SCE and PM. Additionally, the gap between the extensive amount of industrial advice on cost estimation and the low extent of the use of SCE methodologies is addressed. An enhanced understanding of the reasons behind these differences may help organisations improve their SCE success, positively affecting project success.

4. Case contexts

The topics covered in this paper have not been widely addressed prior to this study and our goal was to collect widely different perspectives

Table 3. Case study companies and projects

Company	Software Vendor	Service Provider	Tech Giant
Number of employees	Approximately 150	Several thousands	Several thousands
Business area	Software and services	Software and services	Software and services
Project	Tool	Operational Control System	Network Management System
Initial/actual size of the project	12/44 person-months	20/20 person-months	Approx. 200/200 person-months
Initial/actual duration of the project	3/11 months	10/10 months	3/3 months
Project type	Internal product development	External product development, i.e. tailored software	Continuous internal product development
Estimation methodology	WBS and expert estimation	WBS and expert estimation, historical data, peer review	WBS and expert estimation, historical data
Estimation responsible	Project Manager	Project Manager	Program Manager
Development methodology	Scrumbut: Waterfall (design) + Scrum (sprints)	Waterfall-like method	Scrum
Result	Challenged	Successful	Successful

related to the organisational phenomena affecting SCE, and especially top management's role. Thus, the cases were selected in such a way that they would generate rich information about the phenomena being studied. The authors focused on large and small companies, selecting higher and lower maturity organisations and exemplary and challenged projects. The case companies and projects are different in their industrial domains, size, as well as in their processes. The final decision of including a particular project in the study was made based on a discussion with a company representative, confirming that the project was likely to add new perspectives in the study. Table 3 depicts the characteristics of the case study companies and the projects. The companies wished to remain anonymous.

4.1. Case 1 – Software Vendor's Tool project

Software Vendor is a software producing company of about one hundred and fifty people. Its main line of business consists of selling consultancy and support services as well as software products to businesses. The company is global and has offices in several countries. In this study the Software Vendor's Tool project, which aimed to produce an application development tool, was analysed.

While the overall project was strictly planned beforehand, the actual development work was divided into sprints. The development work started with a prototype version in which technical challenges were studied. The Product Owner and Project Manager were named to the project already in the prototype phase. The Product Owner was responsible for creating a design document for the product, whereas the Project Manager, based on the design document, was responsible for crafting a timetable and cost estimates. Initially, the project was designed to take three months with a team of four people. Based on the estimate and design document, top management approved and started the project.

The Tool project overran its schedule and budget by over 200%. However, the project delivered the planned scope and the Senior Business Manager reports that the outcome of the project met his expectations and he attributes the overruns to estimation error and project performance related issues.

4.2. Case 2 – service provider's operational control system project

Service Provider is a large software producing company with thousands of employees, providing tailor-made and package software, and consul-

tancy services for businesses in various sectors. The company has premises in several countries. For the purpose of this research the Operational Control System project by Service Provider that aims to produce custom software for a long-term customer was studied. The Operational Control System is used for reporting and analysing process control data.

The project followed a Waterfall-like software development process. The first stage of the project was requirement elicitation and analysis. After the specification was approved, the project was estimated. The estimation was made by developers and testers, led by the project manager, who had the overall responsibility of the cost estimate. The estimate was a result of expert estimates, placed into a software tool specifically tailored for the application area.

The project was planned according to certain restrictions: the budget and the timetable was fixed. The development started when the customer and the vendor had agreed upon the scope. There was a small number of unknown features that needed further elaboration. The development work continued straightforwardly from design through implementation and testing to delivery. The duration and effort of the project was 10 months and 600 man-days, respectively. Regardless of a significant rescoping during the project, it concluded under budget and on schedule with good customer satisfaction.

4.3. Case 3 – Tech Giant's network management system project

Tech Giant is a large company selling products with software to global business-to-business markets. The company has tens of thousands of employees around the world. The Network Management System project of Tech Giant was analysed in this research. The project produced a new release of a tool for managing the network. The Network Management System has been in use for several years.

The project was a part of a continuous development cycle involving just under 100 people. A new release of the system is developed every three months. The development methodology it used was based on Scrum with two week sprints. The development teams were distributed over several locations. The cost estimation was conducted in two phases: firstly, rough planning for the whole three month release in the product management function. Secondly, the backlog items were estimated in the Scrum teams, the main responsible being the program manager. The estimate for the whole release was based on historical data about certain parts and the estimates for those parts were prepared by requirement engineers. The backlog items were estimated by using an expert estimation. The project concluded successfully and delivered over 85% of the planned scope, which is the goal for all releases.

5. Case study design

The question of how the organisational phenomena (RQ1) and specifically the actions of top management (RQ2) affect SCE are investigated through three case studies. Since this study deals with contemporary phenomena in a real-world context – over which the researcher has little or no control – the case studies were chosen as a suitable research approach [59]. This study is exploratory, discovering what is happening, seeking new ideas and generating hypotheses and research areas [60]. The research uses a multiple case study design and replication logic [59]. The richness of the information is maximised by using both exemplary and average organisations as cases [61]. The unit of analysis is a single software cost estimate. The study focuses on the experiences gained during the preparation of the cost estimate and the related software process.

To facilitate the identification of organisational phenomena, it was decided to utilise the concept of maturity. Software process maturity is the extent to which a specific process is explicitly defined, managed, measured, controlled and effective [62]. Paulk et al. [62] argue that maturity implies the potential for growth in capability and indicates both the richness of an organisation's process and the consistency with which it is applied in projects. Furthermore, mature

organisations provide training for processes and the processes are monitored and improved. In general, the concept of maturity measures organisational capability, culture and consistency in a holistic way, thus it can be expected to usefully facilitate the discovery of organisational phenomena. Thus the maturity of SCE and software processes are assessed for this study.

5.1. Instrumentation of SCE maturity

To assess the maturity level of SCE in an organisation, the definition of an ideal SCE procedure was developed, it covered its most important aspects as identified in [13]:

1. The use of an estimation methodology: A clearly defined, established estimation methodology is used to produce the estimate, instead of making presumptions.
2. Proper communication of the estimate: The assumptions, accuracy and intended use of an estimate are communicated as part of the estimate, instead of being presented as a figure lacking further explanation.
3. Planned re-estimation: An estimate is improved systematically when information about the assumptions behind an estimate is increased and updated after the initial estimate.
4. The use of a documented estimation procedure: A documented procedure for producing and communicating an estimate is followed, instead of an ad-hoc procedure.

If the above-mentioned areas of SCE are properly covered, the estimation process should avoid many of the worst pitfalls and the outcome will have a fair chance of being useful for project control. As demonstrated by Lederer and Prasad [63], using guessing or intuition as an estimation methodology is connected to budget and schedule overruns. Also, the accuracy of an estimate increases as a project progresses [64, 65], which encourages the re-estimation and good communication of an estimate. In addition, one poorly estimated aspect can become an *anchor* and may contaminate a whole project's estimate [66, 67]. Furthermore, a documented estimation procedure protects organisations from poor estimation practices and promotes good practices [13]. Standardised procedures have also been found to improve the results in PM [19, 68], specifically in software development [15, 69]. Thus, if an estimate is the result of a rigorous procedure covering the above mentioned aspects, it is more likely to be useful.

5.2. Instrumentation of process maturity

In order to ensure that the relevant phenomena are discovered, the scope of this investigation will be extended outside the actual SCE and assess the maturity of the software processes in the studied organisations by using the Capability Maturity Model (CMM) [62]. The CMM establishes a set of publicly available criteria describing the characteristics of mature organisations. CMM presents the process maturity of an organisation in a scale from 1 (low maturity) to 5 (high maturity). For the CMM assessment the general characterisations of maturity levels presented by Paulk et al. and [62, pp. 9–14] key software process area goals [62, pp. 59–64] are used. Together, the CMM characteristics and goals cover a wide range of process areas, so it is probable that reviewing these items will facilitate the discovery of organisational factors affecting SCE, helping to answer RQ1 and RQ2. While CMM is rather old, it still describes well the relevant properties and mechanisms of an organisation, making it a relevant tool for discovering phenomena in the organisational context.

Higher maturity organisations have been found to perform better in software development [70, 71]. The maturity assessment is also related to process areas rather than to techniques, to what rather than to how, making it agnostic to any specific development methodology. Therefore, the software development and estimation maturities are relevant to the discussion of organisational phenomena. The CMM is also specifically intended to be used for software process assessment and software capability evaluations [62].

The CMM evaluation for the case study companies was made by the researchers during the interviews and documentation review. We would like to point out that we followed good audit-

Table 4. Interviewees and their role in the projects

Software Vendor	Service Provider	Tech Giant
Product Owner (key informant)	Project Manager (key informant)	Program Manager (key informant)
Senior Business Manager	Business Manager	Line Manager
Senior Technology Manager	Testing Manager	Senior Manager
Project Manager	Requirements Engineer	Requirements Engineer
	Software Developer	Head of Product Management
		Head of Programs

ing practices and the main author had over five years of experience of auditing and holds an ISO 9001:2008 Lead Auditor certificate. Therefore, we believe that the CMM requirements conformance evaluations conducted as part of the research are valid and we gained a good overall understanding of an organisation's CMM level, even though the focus was still primarily on SCE. In this study the main interest were SCE related topics and CMM acted only as a facilitating instrument.

5.3. Subject selection

The subject sampling strategy was to interview the management and representatives about other roles related to the case projects. In total 15 people were interviewed in 18 interviews (key informants were interviewed twice), as presented in Table 4. All participants attended interviews voluntarily and anonymously and the collected data is treated confidentially.

5.4. Data collection procedures

The data for this study was collected within seven weeks. The primary data collection methods were semi-structured interviews [60] and a review of documentation. In total 15 people were interviewed and 18 documents reviewed. The documents included typical project documentation, such as cost estimates, project plans, meeting minutes and status reports, to gain a better understanding of the procedures and SCE methods used. The case studies were completed one at a time to allow the reflection and refinement of the research and interview questions [72]. All the interviews (but not key informant interviews) related to a single case study were conducted on the same day, with the exception of one interview for the last case study. Each interview lasted approximately one hour. Each interview day was preceded by a key informant interview day during which background information about the case was collected from a person in a central role in the case study area. The key informant interviews addressed the following topics:

1. Project background, size, status and success.
2. Project team members and their roles.
3. Estimation methodology and success.
4. Software development methodology.
5. Software process maturity, capabilities and track record.

The semi-structured interviews were based on a predefined list of questions. Any interesting facts and observations that were mentioned led to additional questions being asked on that subject. The interview instrument was developed by three researchers and adapted slightly for the individual case studies. All the interviews were conducted by two researchers, who interviewed one subject at a time. The interview instrument is provided in Appendix A, it consists of the following main areas:

1. Introduction.
2. Personal, team and project background.
3. Current state of SCE in the organisation.
4. Experiences of the organisational phenomena affecting SCE.
5. Ending (uncovered topics).

5.5. Data analysis procedures

The primary steps for deriving conclusions from the experiences of the study subjects included 1) semi-structured interviews, which were sound

recorded, 2) collection of documentation, 3) transcription of the interviews, 4) the coding of transcripts and documents, 5) grouping the coded pieces of text, and 6) making conclusions. The NVivo 10 application was used for aiding the process, and special care was taken to maintain a clear chain of evidence. The overall process of analysis was conducted as outlined by [73].

During the coding phase, each interview transcript and collected document was reviewed statement by statement, and statements containing information about organisational factors (RQ1) or top management participation (RQ2) were assigned a code representing the findings category. After that, readily coded main categories were reviewed statement by statement to identify subcategories. The subcategories were also identified from the original transcripts. After a couple of iterations, the subcategories emerged from these two approaches. The performed analysis was of the inductive type, meaning that the patterns and categories of the analysis come from the data, instead of being pre-defined. Themes that were often raised in the interviews were identified and coded. The application used for coding (NVivo 10) maintained the evidence trail from the coded pieces of text back to the documents, transcripts and interviewees automatically. The coding of the texts was primarily conducted by one of the researchers. Another researcher conducted a shorter coding of the data, with fewer iterations, independently, to validate the results of the coding. Any differences were discussed and resolved, and the categorisation was refined. The final categorisation formed a structure for reporting the findings of the study.

After the coding of the data, the coded statements were grouped together to form initial hypothesis, or candidates, for conclusions. The process progressed iteratively, and was, once again, conducted primarily by one of the researchers, while another researcher conducted an independent analysis with fewer iterations to validate and refine the results. After a certain number of iterations, and until the end of analysis, the analysis of the statements was conducted by two researchers together. The other two researchers reviewed and validated the results. During the process of forming a hypothesis, interviewees were asked clarifying or additional questions, where deemed necessary, to resolve any unclarities and to provide additional confidence for the hypothesis. The traceability was secured by marking all statements used for forming the hypothesis with identification codes, enabling back tracing to the coded statements.

In addition to the interview data and documentation, the researchers' memos written during the interviews were used as information sources and as part of the data analysis. The collected project documentation provided mostly background data for the case projects, and to some extent, information regarding top management's participation in different phases of the projects. From the organisational context point of view, the documentation provided some information about the software process and related decision making. The role of the collected documentation was mostly to provide background information and to support statements made by the interviewees.

5.6. Validity procedures

The qualitative case study methodology involves the researchers themselves as the instrument of the research, which poses a risk that the results are biased by the researchers' subjective opinions. More generally speaking, Robson [60] identified three types of threats to validity: reactivity, researcher bias and respondent bias. Reactivity means that the presence of the researcher may influence the study, and particularly the behaviour of the study objects. Researcher bias refers to the preconceptions of the researcher, which may influence how questions are asked and answers are interpreted. Finally, respondent bias originates from the respondents' attitudes towards the research, which may lead, for example to withholding information or giving answers the respondents think the researcher is looking for.

Because of the researcher related threat to validity, a discussion of the effects of the involvement of particular researchers is appropriate [60]. The main author of this article has been involved in professional software development since 1996, in-

cluding companies from start-ups to international giant corporations. Additionally, he has been conducting academic research within the area of SCE since 2012, holds an ISO 9001:2008 Lead Auditor certificate, and has over seven years of experience of quality management system audits. The other authors are from academia, having their main focus in software process, software development methodologies and software economy. Together they have published hundreds of research papers, and used different methodologies extensively in their research, including qualitative case studies.

The reactivity, researcher bias and respondent bias threats to the validity of the study were addressed through six strategies provided by [60]: prolonged involvement, triangulation, peer debriefing, member checking, negative case analysis and audit trail. The summary of the taken countermeasures to negate the validity threats are summarised below:

Prolonged involvement: While the study observations were completed during a short period of time, all the researchers had followed the case study companies for at least two years and were intimately aware of recent developments in the software development methodologies being used. All case organisations had participated in a national research programme, Need4Speed (www.n4s.fi), enabling the confidential sharing of information between the organisations and the researchers.

Data source triangulation: Multiple data sources were used, including interviews with persons in different roles, project documentation and informal observations.

Observer triangulation: Interviews were conducted by two researchers together. This also reduced the strain caused by conducting up to six interviews during one day. Additionally, the interviewees had a short break before each interview, and a longer break in the middle of the day. Important analysis steps were conducted by two researchers independently, and emerging issues were discussed and refined.

Methodological triangulation: The data analysis included qualitative interviews and the analysis of project documentation.

Theory triangulation: Several perspectives were considered for interpreting the results, including the perspectives of the subjects, researchers and other peer group members.

Peer debriefing: Peers, including practitioners and researchers, reviewed the research in different research phases. One research paper based on the conducted research has already been published [17]. The results of this research have been reviewed by the Need4Speed research programme steering group.

Member checking: Interviewees reviewed both transcripts and analysis, providing feedback and commentary.

Negative case analysis: Elements that seemed to contradict the conclusions of the analysis were identified and alternative explanations discussed.

Audit trail: Strict scrutiny was practiced to maintain a clear audit trail from data collection to the final conclusions. All interviews, transcripts, codings and other analysis are archived.

Considering that this study is based on three projects, exploratory of nature, and that the study topic has not been widely explored prior to this study, *generalizability* of results is low. However, the study consists of three case companies and 15 interviewees with different roles, and it provides in-depth findings and detailed information of the study itself. Thus, *transferability* of the study should be fair, although case studies are always coloured by their specific context.

6. Results

The following sections present the findings related to organisational phenomena (RQ1) and top management actions (RQ2) affecting SCE. The findings are divided into four main categories (the role of management, communication, process maturity and attitudes) that were found in the analysis and classification of the results by the authors. Additionally, the main categories are divided into subsections as appropriate. The main observations related to the second research question are located in Section 6.1 whereas the

Table 5. Summary of management role findings

	Case 1	Case 2	Case 3
Company	Software Vendor	Service Provider	Tech Giant
Project	Tool	Operational Control System	Network Management System
Estimate purpose	Ensuring the resources, scope and schedule balance, ensuring the minimum viable scope and fast delivery	Preparing an offer for a customer	Ensuring the resources, scope and schedule balance
Participation in estimation	The project plan containing the estimate studied at a summary level, management not aware of the estimation practices	The estimate reviewed on a summary level, management aware of the estimation practices, the project manager scrutinized the estimate	The estimate reviewed on a summary level, management not aware of the estimation practices, the product owner scrutinized the estimate
Resource provisioning	Estimators had enough time for preparing the estimate, prototypes used for supporting estimation	Estimators had enough time for preparing the estimate	Estimators wished to have more time, prototypes used for supporting estimation
Demonstrated importance	Estimates considered as important, confirmed by interviewees	Estimates considered as important, confirmed by interviewees, importance linked to customer promises	Estimates considered as important, confirmed by interviewees, importance linked to customer promises
Goal setting	Goals perceived as realistic, realism pursued, no support for realism from historical data, clear expectations of the scope and schedule, pressure to fit the estimate to expectations	Goals perceived as realistic, realism pursued, hundreds of annually delivered projects supported realism	Goals perceived as realistic, realism pursued, four annual releases for the same product supported realism
Other	No shared project vision		

sections 6.2–6.3 contribute the first research question.

6.1. Management role

Findings related to the management's role are presented in the following sections. Table 5 summarises the findings.

6.1.1. Estimate visibility and purpose

In Case 1, the Tool project, Senior Business Manager studied the project plan containing the estimate considering the strategic importance of the project to the company. In Case 2, the Operational Control System project, the business manager responsible for the important customer relationship reviewed the estimate. Practically, the visibility of the estimate correlated with the ownership of the project and the daily involvement of the managers with the project domain. There was no visibility of the estimate beyond the review as the project was no longer part of the manager's daily responsibilities. In Case 3, the Network Management System project, the most senior manager aware of the estimate was the manager of the whole product family. There are roughly 1,000 experts involved in the system development, so the estimate was visible to relatively senior managers.

In Case 2, the estimate was used for preparing an offer for a customer and planning the project, while in Case 1 and Case 3, the managers reported that they needed the estimate to ensure

that the resources, scope and schedule were in balance with each other. In Case 1, the Senior Business Manager reported that the estimate was needed to ensure the project scope was the minimum viable and that the project would deliver the results as soon as possible.

6.1.2. Participation in estimation

None of the managers studied the estimate in detail. In Case 1, the Senior Business Manager reviewed the estimate only as part of the project plan. In Case 2 and Case 3, the managers reviewed the estimates on a summary level. None of the managers participated in the estimation work, and the managers in Case 1 and Case 3 were not aware of the estimation practices. In Case 2, the manager was aware of the practices because cooperation with the customer was said to be very intense; the customer wanted to discuss processes related to daily cooperation. While the managers were not involved in estimation on a practical level, the managers in cases 2 and 3 stated that they challenged the estimate when necessary. Also, in these two cases, the Project Manager and Product Owner, respectively, scrutinized the estimate. An awareness of such scrutinizing allowed the managers to have greater trust in the estimate. That is, there was no need for them to personally study the estimate in detail.

6.1.3. Resource provisioning

In Case 1 and Case 2, the Tool and Operational Control System projects, the estimators reported that they had enough time to prepare the estimates. In Case 3, the Network Management System project, the estimators wished to have more time. However, although the estimation work was very time consuming and complex, when considering the previous good results, the time reserved for estimation seems to have been reasonable. The perceived lack of time was connected to the complexity and size of the estimation domain. Also, an estimator in Case 3 wondered whether additional time would actually improve the estimates. In Case 1 and Case 3, building prototypes was also used as a method for acquiring additional information to use for estimation, which supported the idea that management provided adequate resources for the estimation work.

6.1.4. Demonstrated importance

In all cases the projects had strong support from management, and the managers emphasized the importance of the estimates. In Case 2 and Case 3, the estimate was strongly linked to keeping the promises given to customers. All the interviewees concurred that management considered the estimates to be of high importance.

6.1.5. Goal setting

All interviewees reported that the project goals seemed realistic and achievable at the beginning of the project, and that everybody pursued realistic estimates. In Case 2, Service Provider delivers hundreds of projects yearly, while in Case 3, Network Management System has four releases per year, thus its management is likely to have a realistic picture of its organisational performance. This probably also supports the setting of realistic and achievable goals for releases and projects. In Case 1, the Tool project was using a new development methodology for the first time, meaning relevant historical data about the process performance was lacking and goal setting was unsupported.

In Case 1, Senior Business Manager expressed the strategic importance of the project, which he had initiated personally, prior to the estimation. Also a roadmap vision, which presented a release date, had been communicated for the product. Furthermore, the scope of the project was considered to be the minimum viable, meaning that the scope could not be reduced. As a result, the estimator was facing a situation in which both the scope and schedule were effectively set, which is always a challenging situation from project planning point of view. The estimator describes having perceived pressure to fit the estimate to these expectations and having started to doubt the estimates when they did not match initial expectations. Case 1, the Tool project, thus seems to have experienced the anchoring phenomena

[66, 67], i.e. the estimate is affected by an expressed starting point. However, Senior Business Manager of Case 1 points out that flexibility in resources and schedule was emphasised prior to estimation.

6.1.6. Provided direction

The interviewees in Case 1 report that there were different expectations for its outcome: Senior Business Manager expected a strong commercial product, while others were building a pre-version, which would contain the full scope of features but not on the quality level expected of a commercial product. The expectation of the rest of the team was that the quality issue would have to be addressed in the next version of the product. This difference in the expectations was probably a significant source of estimation error. Actions for error detection and customer feedback collection add to the amount of work required, as do fixing bugs and improving functionalities based on customer feedback.

6.2. Communication

The role of the written documents, as required by the processes, was significant in Case 1 and Case 2, which followed Waterfall-like development methods. The projects had significantly invested in preparing the documents on which the estimates were heavily reliant. Interviewees from both projects reported that the documents were detailed and of high quality. Also the Network Management System team in Case 3 used documentation as part of its estimation but – as is typical of agile development – it did not have an official role. The documents were prepared on demand when necessary, including pre-studies, memos, presentations and user stories. In addition to the documents, Software Vendor in Case 1 had developed a prototype to get more information on the application area. Prototypes are artefacts, which are likely to support successful estimation because they contain significant amounts of relevant information on the estimated application area and answer many questions relevant to estimation [74]. Tech Giant in Case 3 also reports that it occasionally uses prototypes, while the Business Manager from Service Provider adds that prototypes would be useful but are not utilised at the moment.

While the interviewees recognised the importance of the written documents, all the interviewees in Case 2 and Case 3 emphasised that the process of preparing an estimate is more important than the result itself. The Requirements Engineer and the Project Manager in Case 3 describe the importance of mutual understanding, and all reported that truly understanding each other's needs is crucial. The Requirements Engineer pointed out that estimates become ever more reliable through discussions and said that he is satisfied when all the questions are answered. The Requirements Engineer also highlighted the fact that working together provides confidence in each other. Group estimation sessions were used regularly in both Case 2 and Case 3. The Senior Manager in Case 3 concluded that a good estimate is based on good skills in preparing the specifications and having a broad knowledge about the application area and software development – the majority of the Network Management System project team members in Case 3 had worked on the product for five or more years. Communication seems to be central to estimation in Case 3 because issues like multiple locations and time zones hindering estimation were mentioned. Agile grooming was also mentioned as an important forum for estimation and related communication.

In Case 2, the Project Manager and Testing Manager reported that good cooperation and fact based communication with customers supported estimation. They also emphasised the role of feedback. The interviewees at Case 2 described team members as competent in their area of expertise, stating that estimates were prepared together to a large extent. The Testing Manager added that the atmosphere was open in general. Peer estimation was used on both the programming and PM level. The Project Manager stated that being able to receive consultation or a peer review from another project manager is more important than using information systems to support estimation. The Business Manager added that the project's

estimation succeeded because they understood the customer's needs. The Software Developer expanded on that by saying the estimation succeeded because all the details relevant to the case were found. The Testing Manager described an estimation as meaningful if the right experts were consulted and involved in discussions.

In Case 1, the communication relied more on the documentation. The project manager who prepared the estimate described it as being stored on a shared folder, although no feedback was received. The estimate was based on a design document, which was prepared by the Product Owner. The Project Manager revealed that there had been some discussions with the Product Owner to scope down certain features but the Product Owner and the Senior Technical Manager reported that the estimate had not been challenged at any phase. However, they both stated that they had been sceptical about the estimate but could not point out exactly where the problems resided, and therefore did not raise their reservations. In general, the interviewees reported very few occasions when the estimate would have been discussed. The communication relied mostly on documents prepared by individuals. However, the Senior Technical Manager and Product Owner reported that the atmosphere was open and there was no pressure not to discuss a topic.

6.3. Process maturity

6.3.1. Estimation maturity

All of the case study companies had a documented software process describing how estimation was related to the whole and which documents were required, but only Service Provider in Case 2 had a written procedure for the estimation itself. However, Tech Giant in Case 3 had established estimation procedures, although not documented. Service Provider (Case 2) and Tech Giant (Case 3) had used the same practices for several years, whereas this was the first time for Software Vendor (Case 1) using the estimation procedure in question. The interviewees at Tech Giant and Service Provider reported that they had a history of making successful estimates, while the interviewees at Software Vendor stated that they tend to underestimate and have a poor track record in estimation.

The progress of the project was monitored from the estimation point of view in all case projects. In Case 1, the estimate was presented as a single point estimate. In Case 2, the estimate was presented as a range, consisting of an optimistic, pessimistic and nominal scenario. In Case 3, the target was to deliver at least 85% of the nominal estimate, which can also be seen as a range. The actual project team was more or less known in all projects at the time of estimation. The interviewees in Cases 2 and 3 report that the general estimation capabilities are good, emphasising the importance of professional competence in estimation. The interviewees in Case 1 reported that their estimation capabilities and experience are low. There has also been training related to estimation practices in Case 2 and Case 3. In Case 2, at Service Provider, there was a named person who was responsible for developing estimation practices, which was not the case at the other two companies.

Applying the CMM scale from 1 (low maturity) to 5 (high maturity) and related behavioural characteristics [62, pp. 9–14] to SCE maturity, Service Provider (Case 2) was assessed as being on the highest level, level 5. Their estimation procedures produce reliable results, which are adjusted to specific application areas and technologies and there is systematic work to improve estimation practices. According to our assessment, Tech Giant (Case 3) is on level 4, meaning that while there is room for improvement, the standard processes are defined and established and produce reliable results. Finally, Software Vendor (Case 1) is on level 2, meaning that the processes are defined and may support the production of consistent results. However, in practice, the process discipline was low and the defined practices cannot be applied in real-world situations consistently and successfully.

Table 6 summarises the findings on the SCE procedures used in our case projects; categorised according to the SCE capability criteria defined in Section 4.2. The SCE maturity, when set against

Table 6. Summary of SCE capability findings

	Case 1	Case 2	Case 3
Company Project	Software Vendor Tool	Service Provider Operational Control System	Tech Giant Network Management System
Use of an estimation methodology	(−) No defined standard practice	(+) Work break-down, historical data, software tool	(+) Agile grooming, work break-down, historical data
Proper communication	(+) Assumptions presented (−) Single point	(+) Assumptions presented, range	(+) Assumptions presented, range
Re-estimation and follow-up	(+) Regular follow-up	(+) Regular follow-up	(+) Regular follow-up
Documented estimation procedure	(−) No documented or established procedure	(+) Documented procedure adjusted for the application area, improved continuously	(+) Established, but (−) Not documented
Other	(−) Short experience, low competence, poor track record	(+) Long experience, high competence, good track record	(+) Long experience, high competence, good track record

the criteria in Table 5, seems to correlate well with the CMM maturity levels and the related behavioural characteristics: Service Provider and Tech Giant have practices in place for repeating processes and gaining predictable results. This issue will be discussed more in Section 6.1. There was no standard practices that support the development of consistency at Software Vendor.

6.3.2. Software process maturity

In Case 1, the process used for Tool was relatively new, implemented in the first half of 2014, and was followed by an organisational change in the second half of 2014. The company was adopting Scrum methodology and abandoning the process used in the case project. The Senior Technical Manager of the company said that the primary focus has always been on programming at the cost of other things, such as leadership and PM. The interviewees also referred to similar overruns in projects resembling Tool.

In Case 2, the project manager reported that they deliver hundreds of projects yearly using the same delivery process as used in the case project. The processes are stable and under constant development. According to the Project Manager and Business Manager, the results have been generally good, which was also true of the case project. There was also a training related to the different aspects of the software project delivery model.

Also, Tech Giant in Case 3 has used the current Scrum based process for approximately seven years. According to the Line Manager, the process was under constant development, which was supported by comments from other interviewees. However, the two representatives from product management report that there is still much room for improvement, especially regarding the basing of estimates on current data instead of historical data and the managing of dependencies. Regardless of the pointers for improvement, the product management representative, and other interviewees, described the overall software development performance as good.

To recapitulate, according to our assessment of the overall software process maturity, Software Vendor (Case 1), Service Provider (Case 2) and Tech Giant (Case 3) are on the CMM levels 2, 5 and 4, respectively. A summary of the assessment findings is presented in Appendix B.

6.3.3. Attitudes

All the interviewees in this study recognised the importance of estimation. The reasons for the

experienced importance varied. In Case 3, the Senior Manager argued that estimation facilitates the planning process before the actual work, connecting work to reality. In Case 1, the Project Manager stated that estimation is important from the planning perspective and the Testing Manager in Case 2 concurred. Nevertheless, estimation was experienced as a high importance one. In all case projects, the project manager had the overall responsibility for preparing the estimate. All of the project managers reported that their commitment to the estimate was high.

In Case 1, the general attitudes towards estimation were negative. For example the Senior Technical Manager, Project Manager and Product Owner argued that estimates were not trusted because they were likely to fail. The Senior Technical Manager stated that people were indifferent to the estimates because the usual reaction to overruns was just to continue the project. The Project Manager reported that he did not like giving an estimate and was afraid that the estimate would be interpreted as a commitment. During the re-estimation of the functionalities, the Project Manager described having given upper-bound estimates due to the high level of uncertainty, which also led to the implementation team's reluctance to estimate.

In Case 2, the Customer Manager describes the general attitude towards estimation as good and all the other interviewees agreed, reporting that estimation was a meaningful and motivating task. However, the Testing Manager and Software Developer report that when they are asked for quick and rough estimates, the work does not feel meaningful. They felt that some experts in their company, at Service Provider, take estimation too lightly, not necessarily recognising it as demanding and important work, although the importance of an estimate is understood by all. The Project Manager commented that estimates are sometimes given reluctantly because they are then interpreted as commitments. The Requirements Engineer reported that estimation was not necessarily a pleasant task due to its difficulty. However, the interviewees agreed that estimation generally worked well.

In Case 3, the Requirements Engineer and Project Manager stated that estimation was not a pleasant task, though the discussions are seen as meaningful and relevant. Like the two interviewees in the Operational Control System project, the Requirement Engineer in the Network Management System project said making quick, rough estimates was not motivating. The Line Manager noted that estimators may be afraid that the estimates may not be as desired or that inaccurate estimates will lead to re-planning and corrective actions in the later phases of a project. Estimating was seen as an onerous responsibility. The Senior Manager commented that the development organisation should improve their estimation practices in order to improve the accuracy.

7. Discussion

The following Section 7.1 presents the key findings of this study. The remainder of this section will present the academic (Section 7.2) and practical implications (Section 7.3) of this study, addressing the study's limitations and giving pointers for future research (Section 7.4).

7.1. Key findings

This study focused on gaining insight into top management's role in SCE and discovering organisational phenomena that either support or hinder successful SCE. There were two main research questions: (RQ1) What are the real-world organisational factors that either support or hinder the creation of a meaningful software cost estimate? (RQ2) What is the impact of top management in either supporting or hindering software cost estimation practices?

The primary findings of the study are summarised in Table 7. It was demonstrated that communication, attitudes and process maturity seem to support and hinder the creation of meaningful SCE (RQ1). Furthermore, top management's support and realism were found to support the results of SCE, although anchoring and the lack of a shared project vi-

Table 7. Summary of findings from the case projects by category

	Case 1	Case 2	Case 3
Company	Software Vendor	Service Provider	Tech Giant
Project	Tool	Operational Control System	Network Management System
Outcome	Challenged	Success	Success
Management role	(+) Strong support, realism pursued, enough resources (−) Anchoring, no shared project vision	(+) Strong support, realism pursued, enough resources	(+) Strong support, realism pursued, enough resources
Communication	(+) Detailed plans and specifications, prototype (−) Estimate prepared by one person, lack of discussions and cooperation	(+) Detailed plans and specifications, mutual understanding and insight pursued, cooperation intensive process, expertise and competence emphasised, shared project vision	(+) Aide memoir documentation, mutual understanding and insight pursued, cooperation intensive process, expertise and competence emphasised, shared project vision
Process maturity	(+) Documented software process, regular follow-up (−) No documented estimation procedure, non-established processes, no continuous improvement, no training arranged, low estimation experience and competence, no historical data used	(+) Documented software process, documented estimation procedure, established processes, continuous improvement, training, historical success, high estimation experience and competence, estimate as a range, regular follow-up	(+) Documented software process, established processes, continuous improvement, training, historical success, high estimation experience and competence, estimate as a range, regular follow-up (−) No documented estimation procedure
Attitudes	(+) Importance recognised (+) Project manager commitment high (−) Generally not pleasant, generally negative attitudes, indifference to failure, reluctance	(+) Importance recognised, estimation regarded as meaningful and motivating, general opinion that estimation works well (+) Project manager commitment high (−) Quick, rough estimates not motivating, sometimes unpleasant because of difficulty, some people do not recognise its seriousness, estimates interpreted as commitments	(+) Importance recognised, discussions regarded as meaningful and motivating, general opinion that estimation works well (+) Project manager commitment high (−) Generally not pleasant, quick, rough estimates not motivating, estimates interpreted as commitments, fear of failure, some reluctance

sion were found to hinder SCE (RQ2). Finally, many of the factors affecting SCE, such as communication, providing resources and shared vision, have been found to affect project execution as well. This overlap is natural, since both SCE and project execution are inseparable parts of a software project. Our study, however, focuses on SCE influences, and presents evidence on factors affecting SCE specifically.

7.2. Implications for theory

It has been argued that only a very few papers examine the organisational context of SCE and how its methodologies are applied in real-world situations [14]. According to Jørgensen and Shepperd [14], the basic problems experienced by software companies in relation to SCE are not technical. Hence, this paper has specifically focused on the organisational context related to

SCE and in increasing the understanding of the prerequisites for meaningful SCE. This paper also demonstrates that SCE research remains focused on technical issues, while the focus of PM research has undergone a major shift from a technical to a managerial focus.

The primary finding of this study is that there seems to be a connection between the software process maturity, estimation maturity and estimation success. The maturity as a construct consists of several factors. This study did not identify individual significant organisational factors, which alone would make estimation successful. The connection between the maturity and estimation success suggests that successful estimation is a sum of several factors, such as communication, competence, experience and attitudes.

The more specific results from this study show that commonly used estimation techniques, WBS and expert estimation, can produce good results, if the overall project management and software practices are established and produce consistent results. This paper also suggests that communication is an important factor in the scope of SCE. Furthermore, the findings suggest that SCE should not set any specific requirements for top management, other than that they should carry out their basic responsibilities effectively and avoid the harmful anchoring of estimates.

The finding of this study also correlate well with the previous studies in the area of organisational context and human factors. From the organisational context point of view, Magazinovic and Pernstål [10] researched causes for estimation error, also validating results of Lederer and Prasad's [75] earlier study. They found that management goals affect the results of estimation. This seemed to happen also in Case 1 of this study. Also, in the same study, they found that unclear requirements are a source for estimation error, and that organisations do not have guidelines for conducting cost estimation. Case 1 suffered from unclear requirements, and Case 1 and Case 3 did not have guidelines for estimation. Furthermore, Magazinius, Börjesson and Feldt [9] found that personal agenda, management pressure and attempt to avoid re-estimation may affect the estimate. This seemed to be the case also in the Tool project of this study. The promotion of the project [76] may also explain parts of the tight target setting for the Tool project.

Cognitive bias is another non-technical topic related to SCE, which has gained attention recently. While the primary focus here was in the organisational context, it was discovered the presence of anchoring [66] in Case 1. There also seemed to be, at least to some extent, an attitudinal tendency in all cases to find hindrances for estimation outside the respondent's direct influence, corresponding with [77].

Based on the results presented above, this paper supports the assumption that the estimation challenges experienced in companies are not only technical, but are also related to the organisational context, specifically to the project management and software process maturity. Also, easy to use estimation techniques may not be used by chance but because of the fact that these methods require less organisational capabilities for their successful application. These findings, along with similar findings, should justify SCE researchers shifting their research focus from technical topics to managerial and processual ones.

7.3. Implications for practice

This study addressed the top management's role in software cost estimation. In the following, we will discuss the practical advice found in this research. These are categorized into four groups: top management's role, the importance of communication, organization's process maturity and general attitudes towards SCE.

7.3.1. Top management role

This study suggests that by supporting SCE through the basic TMS practices found in this study, demonstrating SCE's importance, reviewing plans, providing resources and ensuring a shared vision and commitment, top management can create an environment for successful SCE. Earlier studies support this conclusion. For example Boonstra [78] has found that the pro-

vision of resources, the establishment of a clear and well defined project framework, communication with the project team, being knowledgeable about a project and using power to resolve conflicts are important behavioural categories for top management. Zwikael [25] has reported similar findings, and concludes that, e.g. an organisational structure that is supportive of a project, communication between the project manager and the organisation and appropriate project manager assignment have a positive impact on project success. However, the previously defined behaviour is likely to be enough only in an environment where management has already created the necessary capabilities and gained the required experience for successful software work.

On the other hand, the results indicate that if there is a lack of a shared vision or a lack of commitment, the negative impacts on SCE can be significant. This finding receives support from earlier studies. White and Fortune [18] report that 'Clear goals/objectives' was the most frequently mentioned success factor for projects. Fortune and White [40] report that 'Clear realistic goals' was the second most cited factor for success. However, clearly expressed expectations may also become harmful anchors and distort SCE, as found in this and other studies [66, 67].

In summary, successful SCE seems not to require any specific actions from top management, if the general maturity of a work environment is good. Thus, it is enough if management performs its role effectively by providing typical TMS behaviour. However, top management should avoid situations in which their expectations could become anchors that negatively affect SCE.

7.3.2. Communication

The results provide evidence that communication related issues are important factors in successful SCE, when work breakdown structure (WBS) and expert estimation are in use. In both of the successful projects, Cases 2 and 3, the interviewees reported that mutual understanding and understanding the requirements were sought by management. Furthermore, there were many opportunities and forums for discussions on the issues. Hence, cooperation was described as good and the expertise as sufficient for reaching an adequate level of understanding.

There are plenty of similar findings from other areas related to the importance of communication. In the scope of project cost management [31], it was found that early interaction with key stakeholders and the establishment of clear lines of communication for sharing professional and project based knowledge are crucial during the inception phases of projects. Furthermore, the significant role of communication in managing the coordination process was addressed by Malone and Crowston [79]. Communication was found to be a common success factor when discussing change in software projects and teams [80] and the best way to build trust in development teams [81]. Communication was also found to make software development more efficient in companies [82] and was shown to be one of the cornerstones of agile development [83]. In the scope of SCE, Jørgensen [77] noted, in a case study, that poor communication skills or team dynamics might have had an impact on the SCE's result in one team.

On a practical level, these findings suggest that project managers, software professionals and other project team members should focus on achieving an understanding of requirements through discussion, instead of focusing on compliance, techniques and documentation.

7.3.3. Process maturity

All of the case projects used easy to implement [84] estimation methodologies, such as WBS, expert estimation and group estimation. The methodologies seem to produce useful results in a mature environment. Established processes and at least moderate maturity seem to be the key to successful application of estimation methodologies. This conclusion also receives support from earlier research. The success of expert estimation has been shown by Jørgensen [85] and studies on the impact of CMM levels on estimation results show that companies who have levels from three to five produce significantly more accurate results than companies on the lower maturity levels

[13, 86, p. 10]. However, although the estimation accuracy and CMM level seem to correlate with each other, we would like to point out that there seems to be no significant correlation between the project management maturity (PMM) of an organisation and the project success [87]. The correlation between the CMM level and estimation accuracy observed in this study occurs within the studied area of maturity, SCE being part of the software process maturity.

Maturity as a construct consists of several factors, like experience, skills and processes. While we report several maturity related findings connected to successful estimation, like training, experience and continuous improvement, we believe that none of the individual factors is likely to lead to success on its own. However, a lack of one of those factors may have significant negative impacts. Thus, based on our findings, we decided to focus on maturity as a whole, instead of individual factors.

Software process maturity (or project management maturity), estimation maturity and attitudes seem to have a clear interrelationship. If software process maturity is good, estimation maturity seems to be good, furthermore attitudes become more positive. This is not surprising, because SCE is part of a software project and managed under the relevant software project or software process management. The CMM model does not include attitudes in its attributes, although, for example, [88] suggest attitudes are an important factor in project management maturity, in addition to knowledge and action. However, the true relationship between these three is beyond the scope of this study.

Considering the previous and the findings presented in Table 7, it seems intuitive that the overall maturity correlates with the estimation success. This is supported by Flowe and Thordahl [86] and findings from Boeing, presented by McConnell [13, p.10]. Furthermore, each of the elements of maturity is likely to contribute to estimation success also individually. For example Jørgensen [85] has provided evidence that training opportunities, good estimator competence and use of an estimation checklist improve estimation success. In other words, the more there are elements of high maturity present, the higher is the probability of estimation success, and vice versa, low presence of high maturity elements increases uncertainties in estimation.

Our advice for organisations would be to include a simple maturity self-assessment in the software cost estimation process, for example based on a publicly available criteria like CMM or CMMI. If the maturity is assessed to be low, a thorough uncertainty analysis is appropriate. Even the knowledge of high level of uncertainty may help managers in their decision making, even though the uncertainties could not be mitigated. Also, we understand that self-assessments are perhaps not typical for low maturity organisations. However, the use of a simple maturity assessment is far easier than accounting the whole industrial and scientific body of knowledge as individual items. In the beginning, the awareness of the high level of uncertainty could help to make better decisions, and in the longer term act as a list of development pointers towards higher maturity.

For the practitioners in higher maturity organisations, it would be recommended to address specific estimation challenges, like estimating change requests or estimating testing. For example, those two areas seem to be sources of errors [11] and serve to decrease motivation, even in exemplary organisations. Also the relationship between the estimate, target and commitment is not always clear, which was reported as resulting in a reluctance to make estimates; the importance of making a distinction between these three aspects is addressed by McConnell [13].

7.3.4. Attitudes

In cases 2 and 3, project managers had the overall responsibility of preparing the estimate, while the actual estimation was done by software developers. In both projects the estimation was seen as an important and relevant task, and the project managers reported that they were committed to the estimates.

However, in both projects the developers' attitudes towards estimation were negative. Estimation was not considered as a pleasant task and reluctance and low motivation were reported, es-

pecially originating from lack of trust and quickly emerging needs requiring flexibility. Negative attitudes, low motivation and reluctance have been found to decrease the quality of work [89]. Although estimates and outcomes have correlated well in these two projects, it is likely that the risk of estimation error increases when negative attitudes are present, especially in low maturity organisations. Trust and flexibility as values have been found to have a positive effect on project outcome [90]. A trivial advice is to support a positive atmosphere around estimation. However, further research is needed to provide better and more specific advice on this topic.

7.4. Limitations and future work

Although a number of countermeasures to validity threats were taken (see Section 5.6) and the transferability of the results was improved by collecting a rich set of data, this research has certain limitations. This research considered the organisational phenomena at a general level, without taking the project or organisation specific characteristics, like development methodology or company size, into account in the study design.

The findings provide evidence that, at a general level, organisational issues, like the role of management, process maturity and communication, are important factors in SCE. However, although we believe that the results are transferable to similar project settings, the organisational challenges may vary between different contexts. For example, some organisational properties or mechanisms may have been overlooked, such as the size of the company, which causes variation between projects. In addition, there are different reasons for the cost estimates: one company was using them to set the price to the customer while the others were seeking balancing content and timing of their products with the estimates.

Therefore, we encourage further studies in different project and company contexts to see if the same phenomena are repeated, or if there are other context specific phenomena not discovered in this study. Quantitative studies would also provide insight in how commonly the reported phenomena repeat in organisations.

This study also provides evidence that there is an interrelationship between the estimation maturity and project management maturity. This is an important observation, and should be confirmed with a quantitative study that considers a large number of projects as well as studied qualitatively to understand the phenomenon. For example, it might just be that companies with a low CMM level do not recognize that there are situations when it is inappropriate to estimate at all (e.g., new development and estimation methods, new product with no client). This is a lack of risk management procedures, not just an estimation problem.

The findings of this paper are based on three projects, and do not provide a generalizable level of confidence for their relationship. The SCE maturity and software process maturity were also assessed only to the extent necessary for the purposes of this study. We suggest that further studies establish a more precise model for assessing SCE maturity and conduct the actual maturity assessment with maturity as the sole focus of the study.

As an exploratory study, the purpose was also to generate new theories and pointers for further research. One interesting observation revealed by this study was that the attitudes towards estimation were negative among the developers participating in estimation, whereas the attitudes of the project managers were positive and the level of commitment to the estimation high. Negative attitudes may be a source of estimation errors, and increase the probability of overruns. This should be studied further, since negative attitudes hinder any work.

From the construct point of view, the aim was to discover organisational factors affecting SCE. We covered many relevant aspects related to the organisational context in which the estimation took place. Thus, we studied what we planned to study and felt that we developed a clear picture of each of the studied projects.

Generally speaking this study found management and process related topics to be equally important from the SCE point of view as estimation technique related topics. This suggests that SCE research would benefit from approach-

ing those topics from a PM or software process point of view, and that elements from these areas should be synthesised into SCE research. Lastly, as demonstrated in the introduction of this paper, e.g. PM research is more advanced than SCE research on management and other organisation related issues.

8. Conclusions

Many researchers and practitioners argue that organisational issues are equally important from the software estimation success point of view as technical issues. Some of the often cited works related to this important topic have been Lederer and Prasad [75], Jørgensen and Shepperd [14] and Magazinovic and Pernstål [10]. Regardless of this knowledge of the importance of organisational issues in SCE, the focus of the SCE research has remained heavily on estimation methodologies and other technical issues.

The findings of this paper have potential to contribute to the current body of knowledge on organisational issues related to SCE, and specifically on top management's role, in several ways, regardless of the limited transferability of the results. By using the exploratory case study approach and interviewing 15 practitioners involved in software development in three organisations, we have found that the role of top management is important in creating prerequisites for meaningful estimation, but their day-to-day participation is not required for successful estimation. Top management may also induce undesired distortion in estimation. We have also found that estimation maturity and estimation success seem to have an interrelationship with software process maturity, but there seem to be no significant individual organisational factors, which alone would make estimation successful. Additionally, our study validated many of the distortions and biases reported in the earlier studies, and showed that the SCE research focus has remained on estimation methodologies.

Low maturity organisations may be able to reduce overruns through a better understanding of their increased risk level and the existence of good estimation practices. We suggest therefore that future studies and software process improvement activities should pay more attention to low maturity organisations and their specific needs.

Acknowledgements

The authors gratefully acknowledge the support of Tekes – the Finnish Funding Agency for Innovation, DIMECC Oy and the Need for Speed (http://www.n4s.fi) research programme. The funders had no role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

[1] Y.K. Dwivedi, D. Wastell, S. Laumer, H.Z. Henriksen, M.D. Myers, D. Bunker, A. Elbanna, M.N. Ravishankar, and S.C. Srivastava, "Research on information systems failures and successes: Status update and future directions," *Information Systems Frontiers*, Vol. 17, No. 1, 2014, pp. 143–157.

[2] D.L. Hughes, Y.K. Dwivedi, A.C. Simintiras, and N.P. Rana, *Success and Failure of IS/IT Projects.* Springer International Publishing, 2016.

[3] T. Halkjelsvik and M. Jørgensen, "From origami to software development: A review of studies on judgment-based predictions of performance time." *Psychological Bulletin*, Vol. 138, No. 2, 2012, pp. 238–271.

[4] The Standish Group International, "The CHAOS manifesto: Think big and act small," 2013.

[5] D. Galorath, *Software Project Failure Costs Billions. Better Estimation & Planning can Help*, 2012. [Online]. http://galorath.com/blog/software-project-failure-costs-billions-better-estimation-planning-can-help/ [Accessed 16 September 2014].

[6] J. McManus and T. Wood-Harper, *A study in project failure*, 2008. [Online]. http://www.bcs.org/content/ConWebDoc/19584 [Retrieved 19/4/2012].

[7] R.R. Nelson, "IT project management: Infamous failures, classic mistakes, and best practices," *MIS Quarterly Executive*, Vol. 6, No. 2, 2007, pp. 67–78.

[8] M. Jørgensen, "Communication of software cost estimates," in *Proceedings of the 18th International Conference on Evaluation and Assessment*

in Software Engineering – EASE'14. ACM Press, 2014, p. 28.

[9] A. Magazinius, S. Börjesson, and R. Feldt, "Investigating intentional distortions in software cost estimation – An exploratory study," *Journal of Systems and Software*, Vol. 85, No. 8, 2012, pp. 1770–1781.

[10] A. Magazinovic and J. Pernstål, "Any other cost estimation inhibitors?" in *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement – ESEM'08*. ACM Press, 2008, pp. 233–242.

[11] J. Rahikkala, V. Leppänen, J. Ruohonen, and J. Holvitie, "Top management support in software cost estimation," *International Journal of Managing Projects in Business*, Vol. 8, No. 3, 2015, pp. 513–532.

[12] J.J. Ahonen, P. Savolainen, H. Merikoski, and J. Nevalainen, "Reported project management effort, project size, and contract type," *Journal of Systems and Software*, Vol. 109, 2015, pp. 205–213.

[13] S. McConnell, *Software Estimation: Demystifying the Black Art*. Microsoft Press, 2006.

[14] M. Jorgensen and M. Shepperd, "A systematic review of software development cost estimation studies," *IEEE Transactions on Software Engineering*, Vol. 33, No. 1, 2007, pp. 33–53.

[15] B. Pitterman, "Telcordia technologies: The journey to high maturity," *IEEE Software*, Vol. 17, No. 4, 2000, pp. 89–96.

[16] L. Putnam and W. Myers, *Five core metrics: The Intelligence Behind Successful Software Management*. Dorset House Publishing, 2003.

[17] J. Rahikkala, S. Hyrynsalmi, and V. Leppänen, "Accounting testing in software cost estimation: A case study of the current practice and impacts," in *Proceedings of 14th Symposium on Programming Languages and Software Tools*, J. Nummenmaa, O. Sievi-Korte, and E. Mäkinen, Eds. University of Tampere, 2015, pp. 64–75.

[18] D. White and J. Fortune, "Current practice in project management – An empirical study," *International Journal of Project Management*, Vol. 20, No. 1, 2002, pp. 1–11.

[19] *A Guide to the Project Management Body of Knowledge: PMBOK Guide*, Std. ANSI/PMI 99-001-2013, 2013.

[20] N. Cerpa and J.M. Verner, "Why did your project fail?" *Communications of the ACM*, Vol. 52, No. 12, 2009, p. 130.

[21] L. McLeod and S.G. MacDonell, "Factors that affect software systems development project outcomes," *ACM Computing Surveys*, Vol. 43, No. 4, 2011, pp. 1–56.

[22] H.N.N. Mohd and S. Shamsul, "Critical success factors for software projects: A comparative study," *Scientific Research and Essays*, Vol. 6, No. 10, 2011, pp. 2174–2186.

[23] CMMI Product Team, "CMMI for development. version 1.3," Software Engineering Institute, Carnegie Mellon University, Tech. Rep. CMU/SEI-2010-TR-0336, 2010.

[24] G. Caupin, H. Knoepfel, G. Koch, K. Pannenbäcker, F. Pérez-Polo, and C. Seabury, Eds., *IPMA Competence Baseline Version 3*. International Project Management Association, 2006. [Online]. http://www.ipma.world/assets/ICB3.pdf

[25] O. Zwikael, "Top management involvement in project management," *International Journal of Managing Projects in Business*, Vol. 1, No. 4, 2008, pp. 498–511.

[26] T. Cooke-Davies, "The 'real' success factors on projects," *International Journal of Project Management*, Vol. 20, No. 3, 2002, pp. 185–190.

[27] M. Lycett, A. Rassau, and J. Danson, "Programme management: A critical review," *International Journal of Project Management*, Vol. 22, No. 4, 2004, pp. 289–299.

[28] P. Dietrich and P. Lehtonen, "Successful management of strategic intentions through multiple projects – reflections from empirical study," *International Journal of Project Management*, Vol. 23, No. 5, 2005, pp. 386–391.

[29] S. Srivannaboon and D.Z. Milosevic, "A two-way influence between business strategy and project management," *International Journal of Project Management*, Vol. 24, No. 6, 2006, pp. 493–505.

[30] B.J. Kolltveit, J.T. Karlsen, and K. Grønhaug, "Perspectives on project management," *International Journal of Project Management*, Vol. 25, No. 1, 2007, pp. 3–9.

[31] H.K. Doloi, "Understanding stakeholders' perspective of cost estimation in project management," *International Journal of Project Management*, Vol. 29, No. 5, 2011, pp. 622–636.

[32] B. Nanus and L. Farr, "Some cost contributors to large-scale programs," in *Proceedings of the April 21-23, 1964, spring joint computer conference on XX – AFIPS '64 (Spring)*. ACM Press, 1964, pp. 239–248.

[33] E.A. Nelson, "Management handbook for the estimation of computer programming costs," Defense Technical Information Center, Technical Documentary Report ESD-TR-67-66, 1967. [Online]. http://www.dtic.mil/dtic/tr/fulltext/u2/648750.pdf

[34] L.C. Briand and I. Wieczorek, "Resource estimation in software engineering," in *Encyclopedia of*

Software Engineering. John Wiley & Sons, Inc., 2002.

[35] S.K. Sehra, Y.S. Brar, N. Kaur, and S.S. Sehra, "Research patterns and trends in software effort estimation," *Information and Software Technology*, Vol. 91, 2017, pp. 1–21.

[36] J. Hihn and H. Habib-agahi, "Cost estimation of software intensive projects: A survey of current practices," in *Proceedings of the 13th International Conference on Software Engineering*. IEEE Comput. Soc. Press, 1991, pp. 276–287.

[37] R. Turner, *The handbook of project based management*, 2nd ed. McGraw-Hill, 1999.

[38] D.I. Cleland, "The strategic context of projects," in *Project portfolio management – selecting and prioritizing projects for competitive advantage*, L. Dye and J. Pennypacker, Eds. Center for Business Practices, 1999.

[39] L.F. Luna-Reyes, J. Zhang, J. Ramón Gil-García, and A.M. Cresswell, "Information systems development as emergent socio-technical change: A practice approach," *European Journal of Information Systems*, Vol. 14, No. 1, 2005, pp. 93–105.

[40] J. Fortune and D. White, "Framing of project critical success factors by a systems model," *International Journal of Project Management*, Vol. 24, No. 1, 2006, pp. 53–65.

[41] T. Okoro, "Diverse talent: Enhancing gender participation in project management," *Procedia – Social and Behavioral Sciences*, Vol. 226, 2016, pp. 170–175, proceedings of the 29th IPMA World Congress WC2015 (28-30 September–1 October, Panama).

[42] R. Young and E. Jordan, "Top management support: Mantra or necessity?" *International Journal of Project Management*, Vol. 26, No. 7, 2008, pp. 713–725.

[43] R. Schmidt, K. Lyytinen, M. Keil, and P. Cule, "Identifying software project risks: An international Delphi study," *Journal of Management Information Systems*, Vol. 17, No. 4, 2001, pp. 5–36.

[44] M.L. Markus, "Implementation politics: Top management support and user involvement," MIT, Center for Information Systems Research, Alfred P. Sloan School of Management, Tech. Rep. CISR 75, 1981. [Online]. https://dspace.mit.edu/bitstream/handle/1721.1/48186/implementationpo00mark.pdf

[45] Y.H. Kwak and F.T. Anbari, "Analyzing project management research: Perspectives from top management journals," *International Journal of Project Management*, Vol. 27, No. 5, 2009, pp. 435–446.

[46] J. Thomas, C. Delisle, K. Jugdev, and P. Buckle, "Selling project management to senior executives: The case for avoiding crisis sales?" *Project Management Journal*, Vol. 33, No. 2, 2002, pp. 19–28.

[47] L. Crawford, "Senior management perceptions of project management competence," *International Journal of Project Management*, Vol. 23, No. 1, 2005, pp. 7–16.

[48] J.T. Garrity, "Top management and computer profits," *Harvard Business Review*, Vol. 41, No. 4, 1963, pp. 6–12.

[49] C.M. Beath, "Supporting the information technology champion," *MIS Quarterly*, Vol. 15, No. 3, 1991, p. 355.

[50] G.H.A. Morton, "Become a project champion," *International Journal of Project Management*, Vol. 1, No. 4, 1983, pp. 197–203.

[51] *Software Extension to the PMBOK Guide*, 5th ed. Project Management Institute, Inc & IEEE Computer Society, 2013.

[52] S. Adams, *ITIL V3 foundation handbook*. The Stationery Office, 2009, Vol. 1.

[53] *Guidance on Project Management*, Std. 2150:2012, 2012.

[54] M. Cohn, *Agile Estimating and Planning*, ser. Robert C. Martin Series. Pearson Education, Inc., 2005.

[55] M. Petrini and M. Pozzebon, "Managing sustainability with the support of business intelligence: Integrating socio-environmental indicators and organisational context," *The Journal of Strategic Information Systems*, Vol. 18, No. 4, 2009, pp. 178–191.

[56] A. Magazinius and R. Feldt, in *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering – CHASE'10*. ACM Press, 2010, pp. 92–95.

[57] J. Smyrk, "Why most IT projects are really IT without the project," in *Third world project management conference, Gold Coast, Australia*, 2002.

[58] R. Young and E. Jordan, "Lifting the game: Board views on e-commerce risk," in *The Adoption and Diffusion of IT in an Environment of Critical Change*, D. Bunker, D. Wilson, and S. Elliot, Eds., 2002, pp. 102–113.

[59] R.K. Yin, *Case Study Research: Design and Methods*, 3rd ed. SAGE Publications, Inc., 2003.

[60] C. Robson, *Real World Research: A Resource for Social Scientists and Practitioner-researchers*, 2nd ed. Blackwell Publishing, 2002.

[61] M. Patton, *Qualitative Research and Evaluation Method*, 3rd ed. SAGE Publications, Inc., 2001.

[62] M.C. Paulk, B. Curtis, M.B. Chrissis, and C.V. Weber, "Capability maturity

model for software," Software Engineering Institute, Carnegie Mellon University, Tech. Rep. ESC-TR-93-177, 1993. [Online]. https://www.sei.cmu.edu/reports/93tr024.pdf

[63] A.L. Lederer and J. Prasad, "Nine management guidelines for better cost estimating," *Communications of the ACM*, Vol. 35, No. 2, 1992, pp. 51–59.

[64] T. Little, "Schedule estimation and uncertainty surrounding the cone of uncertainty," *IEEE Software*, Vol. 23, No. 3, 2006, pp. 48–54.

[65] B.W. Boehm, R. Madachy, B. Steece *et al.*, *Software cost estimation with Cocomo II*. Prentice Hall PTR, 2000.

[66] M. Jørgensen and D.I. Sjøberg, "The impact of customer expectation on software development effort estimates," *International Journal of Project Management*, Vol. 22, No. 4, 2004, pp. 317–325.

[67] J. Aranda and S. Easterbrook, "Anchoring and adjustment in software estimation," in *Proceedings of the 10th European Software Engineering Conference Held Jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. ACM Press, 2005, pp. 346–355.

[68] D. Milosevic and P. Patanakul, "Standardized project management may increase development projects success," *International Journal of Project Management*, Vol. 23, No. 3, 2005, pp. 181–192.

[69] J.J. Jiang, G. Klein, H.G. Hwang, J. Huang, and S.Y. Hung, "An exploration of the relationship between software development process maturity and project performance," *Information & Management*, Vol. 41, No. 3, 2004, pp. 279–288.

[70] D.R. Goldenson and D.L. Gibson, "Demonstrating the impact and benefits of CMMI^{TM}: An update and preliminary results," Defense Technical Information Center, Tech. Rep., 2003. [Online]. http://www.dtic.mil/dtic/tr/fulltext/u2/a418481.pdf

[71] D.R. Goldenson and J.D. Herbsleb, "After the appraisal: A systematic survey of process improvement, its benefits, and factors that influence success." Defense Technical Information Center, Tech. Rep., 1995. [Online]. http://www.dtic.mil/dtic/tr/fulltext/u2/a300225.pdf

[72] H.K. Klein and M.D. Myers, "A set of principles for conducting and evaluating interpretive field studies in information systems," *MIS Quarterly*, Vol. 23, No. 1, 1999, p. 67.

[73] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empir Software Eng*, Vol. 14, No. 2, 2008, pp. 131–164.

[74] J. Arnowitz, M. Arent, and N. Berger, *Effective Prototyping for Software Makers*. Elsevier, 2007.

[75] A.L. Lederer and J. Prasad, "Causes of inaccurate software development cost estimates," *Journal of Systems and Software*, Vol. 31, No. 2, 1995, pp. 125–134.

[76] A. Magazinius and R. Feldt, "Confirming distortional behaviors in software cost estimation practice," in *37th EUROMICRO Conference on Software Engineering and Advanced Applications*. IEEE, 2011, pp. 411–418.

[77] M. Jørgensen, "Top-down and bottom-up expert estimation of software development effort," *Information and Software Technology*, Vol. 46, No. 1, 2004, pp. 3–16.

[78] A. Boonstra, "How do top managers support strategic information system projects and why do they sometimes withhold this support?" *International Journal of Project Management*, Vol. 31, No. 4, 2013, pp. 498–512.

[79] T.W. Malone and K. Crowston, "The interdisciplinary study of coordination," *ACM Computing Surveys*, Vol. 26, No. 1, 1994, pp. 87–119.

[80] D. Stelzer and W. Mellis, *Software Process: Improvement and Practice*, Vol. 4, No. 4, 1998, pp. 227–250.

[81] K. Henttonen and K. Blomqvist, "Managing distance in a global virtual team: The evolution of trust through technology-mediated relational communication," *Strategic Change*, Vol. 14, No. 2, 2005, pp. 107–119.

[82] M. Paasivaara and C. Lassenius, *Software Process: Improvement and Practice*, Vol. 8, No. 4, 2003, pp. 183–199.

[83] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries *et al.*, "Manifesto for Agile software development," http://www.agilemanifesto.org, 2001.

[84] M. Jorgensen, "Practical guidelines for expert-judgment-based software effort estimation," *IEEE Software*, Vol. 22, No. 3, 2005, pp. 57–63.

[85] M. Jørgensen, "A review of studies on expert estimation of software development effort," *Journal of Systems and Software*, Vol. 70, No. 1-2, 2004, pp. 37–60.

[86] R.M. Flowe and J.B. Thordahl, "A correlational study of the SEI's capability maturity model and software development performance in DoD contracts." DTIC Document, Tech. Rep., 1994.

[Online]. http://www.dtic.mil/dtic/tr/fulltext/u2/a288890.pdf

[87] H.J. Yazici, "The role of project management maturity and organizational culture in perceived performance," *Project Management Journal*, Vol. 40, No. 3, 2009, pp. 14–33.

[88] E.S. Andersen and S.A. Jessen, "Project maturity in organisations," *International Journal of Project Management*, Vol. 21, No. 6, 2003, pp. 457–461.

[89] A. Raja, W. Mohsin, N. Ehsan, E. Mirza, and M. Saud, "Impact of emotional intelligence and work attitude on quality of service in the call centre industry of Pakistan," in *IEEE International Conference on Management of Innovation & Technology*. IEEE, 2010, pp. 402–407.

[90] Y.J.T. Zidane, B.A. Hussein, J.Ø. Gudmundsson, and A. Ekambaram, "Categorization of organizational factors and their impact on project performance," *Procedia – Social and Behavioral Sciences*, Vol. 226, 2016, pp. 162–169.

Appendix A. Interview instrument

1. Introduction (approximately 5 minutes):
   - A brief introduction to the study.
   - An introduction of the benefits of participation.
   - Anonymity and confidentiality.
2. Personal, team and project background (approximately 5 minutes):
   - Interviewee's personal history and job position in the company.
   - Background of the estimated project and the development methodology that was used.
3. Current state of SCE in the organisation (approximately 25 minutes):
   - Describe the procedure for creating the estimate.
   - Describe the method for creating the estimate of the effort required.
   - Describe the responsibilities related to maintaining and improving the software and estimation practices.
   - Describe the outcome of the estimation.
   - Describe the approach to re-estimation during the project.
4. Experiences of organisational phenomena affecting the four SCE aspects (approximately 20 minutes):
   - Describe the management, project manager and personal expectations of the estimate.
   - Describe the overall SCE skills and motivation in your organisation during the estimation.
   - Describe the demonstrated importance and attitudes regarding the estimate.
   - Describe the ways in which top management and other stakeholders were involved in SCE.
   - Did the project have clear goals and realistic expectations?
   - Was there pressure to make the estimate smaller or other pressures?
   - Was the estimate allowed to change over time?
   - Was there enough time allocated for preparing the estimate?
   - Did all stakeholders seek realistic and accurate estimates?
   - What was the level of commitment of different stakeholders to the estimate?
   - What were the primary issues hindering and supporting successful estimation?
5. Ending (approximately 5 minutes):
   - Any other relevant observations that we have not covered?

Appendix B. Software process CMM level assessment summary

The following tables B1, B2, B3 and B4 presents our CMM assessments for levels 2, 3, 4 and 5, respectively, to the case study companies.

Table B1. The key process areas for level 2: repeatable

Process area	Goal	Software Vendor	Service Provider	Tech Giant
Requirements Management	System requirements allocated to software are controlled to establish a baseline for software engineering and management use	Yes	Yes	Yes
Requirements Management	Software plans, products, and activities are kept consistent with the system requirements allocated to software	Yes	Yes	Yes
Software Project Planning	Software estimates are documented for use in planning and tracking the software project	Yes	Yes	Yes
Software Project Planning	Software project activities and commitments are planned and documented	Yes	Yes	Yes
Software Project Planning	Affected groups and individuals agree to their commitments related to the software project	N/A	N/A	N/A
Software Project Tracking and Oversight	Actual results and performances are tracked against the software plans	Yes	Yes	Yes
Software Project Tracking and Oversight	Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans	Yes	Yes	Yes
Software Project Tracking and Oversight	Changes to software commitments are agreed to by the affected groups and individuals	N/A	N/A	N/A
Software Subcontract Management	The prime contractor selects qualified software subcontractors	N/A	N/A	N/A
Software Subcontract Management	The prime contractor and the software subcontractor agree to their commitments to each other	N/A	N/A	N/A
Software Subcontract Management	The prime contractor and the software subcontractor maintain ongoing communications	N/A	N/A	N/A
Software Subcontract Management	The prime contractor tracks the software subcontractor's actual results and performance against its commitments	N/A	N/A	N/A
Software Quality Assurance	Software quality assurance activities are planned	Yes	Yes	Yes
Software Quality Assurance	Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively	Yes	Yes	Yes
Software Quality Assurance	Affected groups and individuals are informed of software quality assurance activities and results	Yes	Yes	Yes
Software Quality Assurance	Noncompliance issues that cannot be resolved within the software project are addressed by senior management	N/A	N/A	N/A
Software Network Management Management	Software configuration management activities are planned	N/A	N/A	N/A
Software Network Management Management	Selected software work products are identified, controlled, and available	N/A	N/A	N/A
Software Network Management Management	Changes to identified software work products are controlled	N/A	N/A	N/A
Software Network Management Management	Affected groups and individuals are informed of the status and content of software baselines	N/A	N/A	N/A

Notes: Yes – assessment provides evidence of fulfilling the goal; N/A – fulfillment of the goal was not assessed

Table B2. The key process areas for level 3: defined

Process area	Goal	Software Vendor	Service Provider	Tech Giant
Organization Process Focus	Software process development and improvement activities are coordinated across the organization	No	Yes	Yes
Organization Process Focus	The strengths and weaknesses of the software processes used are identified relative to a process standard	N/A	N/A	N/A
Organization Process Focus	Organization-level process development and improvement activities are planned	No	Yes	Yes
Organization Process Definition	A standard software process for the organization is developed and maintained	Yes	Yes	Yes
Organization Process Definition	Information related to the use of the organization's standard software process by the software projects is collected, reviewed, and made available	N/A	N/A	N/A
Training Program	Training activities are planned	No	Yes	Yes
Training Program	Training for developing the skills and knowledge needed to perform software management and technical roles is provided	No	Yes	Yes
Training Program	Individuals in the software engineering group and software-related groups receive the training necessary to perform their roles	No	Yes	Yes
Integrated Software Management	The project's defined software process is a tailored version of the organization's standard software process	N/A	N/A	N/A
Integrated Software Management	The project is planned and managed according to the project's defined software process	Yes	Yes	Yes
Software Product Engineering	The software engineering tasks are defined, integrated, and consistently performed to produce the software	Yes	Yes	Yes
Software Product Engineering	Software work products are kept consistent with each other	N/A	N/A	N/A
Intergroup Coordination	The customer's requirements are agreed to by all affected groups	No	Yes	Yes
Intergroup Coordination	The commitments between the engineering groups are agreed to by the affected groups	N/A	N/A	N/A
Intergroup Coordination	The engineering groups identify, track, and resolve intergroup issues	N/A	N/A	N/A
Peer Reviews	Peer review activities are planned	No	Yes	Yes
Peer Reviews	Defects in the software work products are identified and removed	Yes	Yes	Yes

Notes: Yes – assessment provides evidence of fulfilling the goal; No – assessment provides evidence of not fulfilling the goal; N/A – fulfillment of the goal was not assessed

Table B3. The key process areas for level 4: managed

Process area	Goal	Software Vendor	Service Provider	Tech Giant
Quantitative Process Management	The quantitative process management activities are planned	No	Yes	Yes
Quantitative Process Management	The process performance of the project's defined software process is controlled quantitatively	N/A	N/A	N/A
Quantitative Process Management	The process capability of the organization's standard software process is known in quantitative terms	No	Yes	Yes
Software Quality Management	The project's software quality management activities are planned	Yes	Yes	Yes
Software Quality Management	Measurable goals for software product quality and their priorities are defined	N/A	N/A	N/A
Software Quality Management	Actual progress toward achieving the quality goals for the software products is quantified and managed	N/A	N/A	N/A

Notes: Yes – assessment provides evidence of fulfilling the goal; No – assessment provides evidence of not fulfilling the goal; N/A – fulfillment of the goal was not assessed

Table B4. The key process areas for level 5: optimizing

Process area	Goal	Software Vendor	Service Provider	Tech Giant
Defect Prevention	Defect prevention activities are planned	Yes	Yes	Yes
Defect Prevention	Common causes of defects are sought out and identified	N/A	N/A	Yes
Defect Prevention	Common causes of defects are prioritized and systematically eliminated	N/A	N/A	N/A
Technology Change Management	Incorporation of technology changes are planned	N/A	N/A	N/A
Technology Change Management	New technologies are evaluated to determine their effect on quality and productivity	N/A	N/A	N/A
Technology Change Management	Appropriate new technologies are transferred into normal practice across the organization	N/A	N/A	N/A
Process Change Management	Continuous process improvement is planned	No	Yes	Yes
Process Change Management	Participation in the organization's software process improvement activities is organization wide	No	Yes	Yes
Process Change Management	The organization's standard software process and the projects' defined software processes are improved continuously	No	Yes	Yes

Notes: Yes – assessment provides evidence of fulfilling the goal; No – assessment provides evidence of not fulfilling the goal; N/A – fulfillment of the goal was not assessed

e-Informatica Software Engineering Journal, Volume 12, Issue 1, 2018, pages: 199–216, DOI 10.5277/e-Inf180108

Applying Machine Learning to Software Fault Prediction

Bartłomiej Wójcicki*, Robert Dąbrowski*
*Institute of Informatics, University of Warsaw
bartwojcicki@gmail.com, r.dabrowski@mimuw.edu.pl

Abstract

Introduction: Software engineering continuously suffers from inadequate software testing. The automated prediction of possibly faulty fragments of source code allows developers to focus development efforts on fault-prone fragments first. Fault prediction has been a topic of many studies concentrating on C/C++ and Java programs, with little focus on such programming languages as Python.

Objectives: In this study the authors want to verify whether the type of approach used in former fault prediction studies can be applied to Python. More precisely, the primary objective is conducting preliminary research using simple methods that would support (or contradict) the expectation that predicting faults in Python programs is also feasible. The secondary objective is establishing grounds for more thorough future research and publications, provided promising results are obtained during the preliminary research.

Methods: It has been demonstrated [1] that using machine learning techniques, it is possible to predict faults for C/C++ and Java projects with recall 0.71 and false positive rate 0.25. A similar approach was applied in order to find out if promising results can be obtained for Python projects. The working hypothesis is that choosing Python as a programming language does not significantly alter those results. A preliminary study is conducted and a basic machine learning technique is applied to a few sample Python projects. If these efforts succeed, it will indicate that the selected approach is worth pursuing as it is possible to obtain for Python results similar to the ones obtained for C/C++ and Java. However, if these efforts fail, it will indicate that the selected approach was not appropriate for the selected group of Python projects.

Results: The research demonstrates experimental evidence that fault-prediction methods similar to those developed for C/C++ and Java programs can be successfully applied to Python programs, achieving recall up to 0.64 with false positive rate 0.23 (mean recall 0.53 with false positive rate 0.24). This indicates that more thorough research in this area is worth conducting.

Conclusion: Having obtained promising results using this simple approach, the authors conclude that the research on predicting faults in Python programs using machine learning techniques is worth conducting, natural ways to enhance the future research being: using more sophisticated machine learning techniques, using additional Python-specific features and extended data sets.

Keywords: classifier, fault prediction, machine learning, metric, Naïve Bayes, Python, quality, software intelligence

1. Introduction

Software engineering is concerned with the development and maintenance of software systems. Properly engineered systems are reliable and they satisfy user requirements while at the same time their development and maintenance is affordable.

In the past half-century computer scientists and software engineers have come up with numerous ideas for how to improve the discipline of software engineering. Structural programming [2]

restricted the imperative control flow to hierarchical structures instead of *ad-hoc* jumps. Computer programs written in this style were more readable, easier to understand and reason about. Another improvement was the introduction of an object-oriented paradigm [3] as a formal programming concept.

In the early days software engineers perceived significant similarities between software and civil engineering processes. The waterfall model [4], which resembles engineering practices, was widely adopted as such regardless of its original description actually suggesting a more agile approach.

It has soon turned out that building software differs from building skyscrapers and bridges, and the idea of extreme programming emerged [5], its key points being: keeping the code simple, reviewing it frequently and early and frequent testing. Among numerous techniques, a test-driven development was promoted which eventually resulted in the increased quality of produced software and the stability of the development process [6]. Contemporary development teams started to lean towards short iterations (sprints) rather than fragile upfront designs, and short feedback loops, thus allowing customers' opinions to provide timely influence on software development. This meant creating even more complex software systems.

The growing complexity of software resulted in the need to describe it at different levels of abstraction, and, in addition to this, the notion of software architecture has developed. The emergence of patterns and frameworks had a similar influence on the architecture as design patterns and idioms had on programming. Software started to be developed by assembling reusable software components which interact using well-defined interfaces, while component-oriented frameworks and models provided tools and languages making them suitable for formal architecture design.

However, a discrepancy between the architecture level of abstraction and the programming level of abstraction prevailed. While the programming phase remained focused on generating a code within a preselected (typically object-oriented) programming language, the architecture phase took place in the disconnected component world. The discrepancies typically deepened as the software kept gaining features without being properly refactored, development teams kept changing over time working under time pressure with incomplete documentation and requirements that were subject to frequent changes. Multiple development technologies, programming languages and coding standards made this situation even more severe. The unification of modelling languages failed to become a silver bullet.

The discrepancy accelerated research on software architecture and the automation of software engineering. This includes the vision for the automated engineering of software based on architecture warehouse and software intelligence [7] ideas. The architecture warehouse denotes a repository of the whole software system and software process artefacts. Such a repository uniformly captures and regards as architectural all information which was previously stored separately in design documents, version-control systems or simply in the minds of software developers. Software intelligence denotes a set of tools for the automated analysis, optimization and visualization of the warehouse content [8, 9].

An example of this approach is combining information on source code artefacts, such as functions, with the information on software process artefacts, such as version control comments indicating the developers' intents behind changes in given functions. Such an integration of source code artefacts and software process artefacts allows to aim for more sophisticated automated learning and reasoning in the area of software engineering, for example obtaining an ability to automatically predict where faults are likely to occur in the source code during the software process.

The automated prediction of possibly faulty fragments of the source code, which allows developers to focus development efforts on the bug prone modules first, is the topic of this research. This is an appealing idea since, according to a U.S. National Institute of Standards and Technology's study [10], inadequate software testing infrastructure costs the U.S. economy an esti-

mated $60 billion annually. One of the factors that could yield savings is identifying faults at earlier development stages.

For this reason, fault prediction was the subject of many previous studies. As yet, software researchers have concluded that defect predictors based on machine learning methods are practical [11] and useful [12]. Such studies were usually focused on C/C++ and Java projects [13] omitting other programming languages, such as Python.

This study demonstrates experimentally that techniques used in the former fault prediction studies can be successfully applied to the software developed in Python. The paper is organized as follows: in section 2 the related works are recalled; in section 3 the theoretic foundations and implementation details of the approach being subject of this study are highlighted; the main results are presented in section 4, with conclusions to follow in section 5. The implementation of the method used in this study for predictor evaluation is outlined in the Appendix, it can be used to reproduce the results of the experiments. The last section contains bibliography.

2. Related work

Software engineering is a sub-field of applied computer science that covers the principles and practice of architecting, developing and maintaining software. Fault prediction is a software engineering problem. Artificial intelligence studies software systems that are capable of intelligent reasoning. Machine learning is a part of artificial intelligence dedicated to one of its central problems - automated learning. In this research machine learning methods are applied to a fault prediction problem.

For a given Python software project, the architectural information warehoused in the project repository is used to build tools capable of automated reasoning about possible faults in a given source code. More specifically: (1) a tool able to predict which parts of the source code are fault-prone is developed; and (2) its operation is demonstrated on five open-source projects.

Prior works in this field [1] demonstrated that it is possible to predict faults for C/C++ and Java projects with a recall rate of 71% and a false positive rate of 25%. The tool demonstrated in this paper demonstrates that it is possible to predict faults in Python achieving recall rates up to 64% with a false positive rate of 23% for some projects; for all tested projects the achieved mean recall was 53% with a false positive rate of 24%.

Fault prediction spans multiple aspects of software engineering. On the one hand, it is a software verification problem. In 1989 Boehm [14] defined the goal of verification as an answer to the question *Are we building the product right*? Contrary to formal verification methods (e.g. model checking), fault predictors cannot be used to prove that a program is correct; they can, however, indicate the parts of the software that are suspected of containing defects.

On the other hand, fault prediction is related to software quality management. In 2003 Khoshgoftaar et al. [15] observed that it can be particularly helpful in prioritizing quality assurance efforts. They studied high-assurance and mission-critical software systems heavily dependent on the reliability of software applications. They evaluated the predictive performance of six commonly used fault prediction techniques. Their case studies consisted of software metrics collected over large telecommunication system releases. During their tests it was observed that prediction models based on software metrics could actually predict the number of faults in software modules; additionally, they compared the performance of the assessed prediction models.

Static code attributes have been used for the identification of potentially problematic parts of a source code for a long time. In 1990 Porter et al. [16] addressed the issue of the early identification of high-risk components in the software life cycle. They proposed an approach that derived the models of problematic components based on their measurable attributes and the attributes of their development processes. The models allowed to forecast which components were likely to share the same high-risk properties, such as like being error-prone or having a high development cost.

Table 1. Prior results of fault predictors using NASA data sets [17]

Data set	Language	Recall	False positive rate
PC1	C	0.24	0.25
JM1	C	0.25	0.18
CM1	C	0.35	0.10
KC2	C++	0.45	0.15
KC1	C++	0.50	0.15
	In total:	0.36	0.17

Table 2. Prior results of fault predictors using NASA data sets [1] (logarithmic filter applied)

Data set	Language	Recall	False positive rate
PC1	C	0.48	0.17
MW1	C	0.52	0.15
KC3	Java	0.69	0.28
CM1	C	0.71	0.27
PC2	C	0.72	0.14
KC4	Java	0.79	0.32
PC3	C	0.80	0.35
PC4	C	0.98	0.29
	In total:	0.71	0.25

In 2002, the NASA Metrics Data Program Data sets were published [18]. Each data set contained complexity metrics defined by Halstead and McCabe, the lines of code metrics and defect rates for the modules of a different subsystem of NASA projects. These data sets included projects in C, C++ and Java. Multiple studies that followed used these data sets and significant progress in this area was made.

In 2003 Menzies et al. examined decision trees and rule-based learners [19–21]. They researched a situation when it is impractical to rigorously assess all parts of complex systems and test engineers must use some kind of defect detectors to focus their limited resources. They defined the properties of good defect detectors and assessed different methods of their generation. They based their assessments on static code measures and found that (1) such defect detectors yield results that are stable across many applications, and (2) the detectors are inexpensive to use and can be tuned to the specifics of current business situations. They considered practical situations in which software costs are assessed and additionally assumed that better assessment allowed to earn exponentially more money. They pointed out that given finite budgets, assessment resources are typically skewed towards areas that are believed to be mission critical; hence, the portions of the system that may actually contain defects may be missed. They indicated that by using proper metrics and machine learning algorithms, quality indicators can be found early in the software development process.

In 2004 Menzies et al. [17] assessed other predictors of software defects and demonstrated that these predictors are outperformed by Naïve Bayes classifiers, reporting a mean recall of 0.36 with a false positive rate of 0.17 (see Table 1). More precisely they demonstrated that when learning defect detectors from static code measures, Naïve Bayes learners are better than entropy-based decision-tree learners, and that accuracy is not a useful way to assess these detectors. They also argued that such learners need no more than 200–300 examples to learn adequate detectors, especially when the data has been heavily stratified; i.e. divided into sub-sub-sub systems.

In 2007 Menzies et al. [1] proposed applying a logarithmic filter to features. The value of using static code attributes to learn defect predictors was widely debated. Prior work explored issues such as the merits of McCabes versus Halstead versus the lines of code counts for generating defect predictors. They showed that such debates are irrelevant since *how* the attributes are used to build predictors is much more important than *which* particular attributes are actually used. They demonstrated that adding a logarithmic filter resulted in improving recall to 0.71, keeping a false positive rate reasonably low at 0.25 (see Table 2).

In 2012 Hall et al. [13] identified and analysed 208 defect prediction studies published from January 2000 to December 2010. By a systematic review, they drew the following conclusions: (1) there are multiple types of features that can be used for defect prediction, including static code metrics, change metrics and previous fault

metrics; (2) there are no clear best bug-proneness indicators; (3) models reporting a categorical predicted variable (e.g. fault prone or not fault prone) are more prevalent than models reporting a continuous predicted variable; (4) various statistical and machine learning methods can be employed to build fault predictors; (5) industrial data can be reliably used, especially data publicly available in the NASA Metrics Data Program data sets; (6) fault predictors are usually developed for C/C++ and Java projects.

In 2016 Lanza et al. [22] criticized the evaluation methods of defect prediction approaches; they claimed that in order to achieve substantial progress in the field of defect prediction (also other types of predictions), researchers should put predictors out into the real world and have them assessed by developers who work on a live code base, as defect prediction only makes sense if it is used *in vivo*.

The main purpose of this research is to extend the range of analysed programming languages to include Python. In the remaining part of the paper it is experimentally demonstrated that it is possible to predict defects for Python projects using static code features with an approach similar to (though not directly replicating) the one taken by Menzies et al. [1] for C/C++ and Java.

3. Problem definition

For the remaining part of this paper let *fault* denote any flaw in the source code that can cause the software to fail to perform its required function. Let *repository* denote the storage location from which the source code may be retrieved with version control capabilities that allow to analyse *revisions* denoting the precisely specified incarnations of the source code at a given point in time. For a given revision K let $K{\sim}1$ denote its parent revision, $K{\sim}2$ denote its grandparent revision, etc. Let *software metric* denote the measure of a degree to which a unit of software possesses some property. Static metrics can be collected for software without executing it, in contrast to the dynamic ones. Let *supervised learning* denote a type of machine learning task where an algorithm learns from a set of training examples with assigned expected outputs [23].

The authors follow with the definition central to the problem researched in this paper.

Definition 3.1. Let a *classification problem* denote an instance of a machine learning problem, where the expected output is *categorical*, that is where: a *classifier* is the algorithm that implements the classification; a *training set* is a set of instances supplied for the classifier to learn from; a *testing set* is a set of instances used for assessing classifier performance; an *instance* is a single object from which the classifier will learn or on which it will be used, usually represented by a *feature vector* with *features* being individual measurable properties of the phenomenon being observed, and a *class* being the predicted variable, that is the output of the classifier for the given instance.

In short: in classification problems classifiers assign classes to instances based on their features.

Fault prediction is a process of predicting where faults are likely to occur in the source code. In this case machine learning algorithms operate on instances being units of code (e.g. functions, classes, packages). Instances are represented by their features being the properties of the source code that indicate the source code unit's fault-proneness (e.g. number of lines of code, number of previous bugs, number of comments). The features are sometimes additionally preprocessed; an example of a feature preprocessor, called a *logarithmic filter*, substitutes the values of features with their logarithms. For the instances in the training set the predicted variable must be provided; e.g. the instances can be reviewed by experts and marked as *fault-prone* or *not fault-prone*. After the fault predictor learns from the training set of code units, it can be used to predict the fault-proneness of the new units of the code. The process is conceptually depicted in Figure 1.

A *confusion matrix* is a matrix containing the counts of instances grouped by the actual and predicted class. For the classification problem it is a 2×2 matrix (as depicted in Table 3). The confusion matrix and derived metrics can be used to evaluate classifier performance, where the typical indicators are as follows:

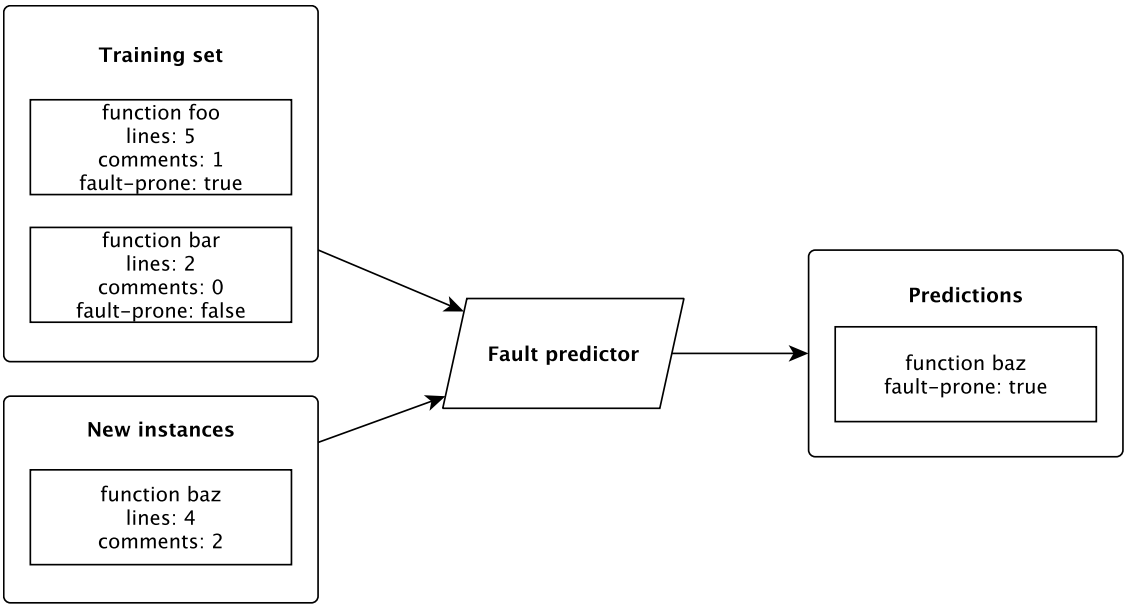


Figure 1. Fault prediction problem (sample)

Table 3. Confusion matrix for classification problems

Actual/predicted	Negative	Positive
negative	true negative (tn)	false positive (fp)
positive	false negative (fn)	true postive (tp)

Definition 3.2. Let *recall* denote a fraction of actual positive class instances that are correctly assigned to positive class:

$$\frac{tp}{tp + fn}$$

Let *precision* denote a fraction of predicted positive class instances that actually are in the positive class:

$$\frac{tp}{tp + fp}$$

Let a *false positive rate* denote a fraction of actual negative class instances that are incorrectly assigned to the positive class:

$$\frac{fp}{fp + tn}$$

Let *accuracy* denote a fraction of instances assigned to correct classes:

$$\frac{tp + tn}{tp + fp + tn + fn}$$

The remaining part of this section contains two subsections. In 3.1 the classification problem analysed in this study is stated in terms typical to machine learning, that is instances: what kinds of objects are classified; classes: into what classes are they are divided; features: what features are used to describe them; classifier: which learning method is used. Section 3.2 focuses on the practical aspects of fault prediction and describes the operational phases of the implementation: identification of instances, feature extraction, generation of a training set, training and predicting.

3.1. Classification problem definition

3.1.1. Instances

The defect predictor described in this study operates at the function level, which is a *de facto* standard in this field [13]. As *the first rule of functions is that they should be small* [24], it

was assumed that it should be relatively easy for developers to find and fix a bug in a function reported as fault-prone by a function-level fault predictor. Hence, in this research functions being instances of problem definition were selected.

3.1.2. Classes

For simplicity of reasoning, in this research the severity of bugs is not predicted. Hence, problem definition instances are labelled as either *fault-prone* or *not fault-prone*.

3.1.3. Features

To establish defect predictors the code complexity measures as defined by McCabe [25] and Halstead [26] were used.

The following Halstead's complexity measures were applied in this study as code metrics for estimating programming effort. They estimate complexity using operator and operand counts and are widely used in fault prediction studies [1].

Definition 3.3. Let n_1 denote the count of distinct operators, n_2 denote the count of distinct operands, N_1 denote the total count of operators, N_2 denote the total count of operands. Then Halstead metrics are defined as follows: program vocabulary $n = n_1 + n_2$; program length $N = N_1 + N_2$; calculated program length $\widehat{N} = n_1 \log_2 n_1 + n_2 \log_2 n_2$; volume $V = N \times \log_2 n$; difficulty $D = n_1/2 \times N_2/n_2$; effort $E = D \times V$; time required to program $T = E/18$ seconds; number of delivered bugs $B = V/3000$.

In this research all the metrics defined above, including the counters of operators and operands, are used as features; in particular preliminary research indicated that limiting the set of features leads to results with lower recall.

In the study also The McCabe's cyclomatic complexity measure, being quantitative measure of the number of linearly independent paths through a program's source code, was applied. In terms of the software's architecture graph, cyclomatic complexity is defined as follows.

Definition 3.4. Let G be the flow graph being a subgraph of the software architecture graph, where e denotes the number of edges in G and n denotes the number of nodes in G. Then cyclomatic complexity CC is defined as $CC(G) = e - n + 2$.

It is worth noting that some researchers oppose using cyclomatic complexity for fault prediction. Fenton and Pfleeger argue that it is highly correlated with the lines of code, thus it carries little information [27]. However, other researchers used McCabe's complexity to build successful fault predictors [1]. Also industry keeps recognizing cyclomatic complexity measure as useful and uses it extensively, as it is straightforward and can be communicated across the different levels of development stakeholders [28]. In this research the latter opinions are followed.

3.1.4. Classifier

In this study, the authors opted for using a Naïve Bayes classifier. Naïve Bayes classifiers are a family of supervised learning algorithms based on applying Bayes' theorem with naïve independence assumption between the features. In preliminary experiments, this classifier achieved significantly higher recall than other classifiers that were preliminary considered. Also, as mentioned in section 2, it achieved best results in previous fault prediction studies [1].

It should be noted that for a class variable y and features $x_1, \ldots, x_n$, Bayes' theorem states the following relationship:

$$P(y|x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots, x_n|y)}{P(x_1, \ldots, x_n)}.$$

This relationship can be simplified using the naïve independence assumption:

$$P(y|x_1, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, \ldots, x_n)}.$$

Since $P(x_1, \ldots, x_n)$ does not depend on y, then the following classification rule can be used:

$$\widehat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i|y),$$

where $P(y)$ and $P(x_i|y)$ can be estimated using the training set. There are multiple variants of the Naïve Bayes classifier; in this paper a Gaus-

sian Naïve Bayes classifier is used which assumes that the likelihood of features is Gaussian.

3.2. Classification problem implementation

3.2.1. Identification of instances

A fault predicting tool must be able to encode a project as a set of examples. The identification of instances is the first step of this process. This tool implements it as follows: (1) it retrieves a list of files in a project from a repository (Git); (2) it limits results to a source code (Python) files; (3) for each file it builds an Abstract Syntax Tree (AST) and walks the tree to find the nodes representing source code units (functions).

3.2.2. Feature extraction

A fault predictor expects instances to be represented by the vectors of features. This tool extracts those features in the following way. Halstead metrics are derived from the counts of operators and operands. To calculate them for a given instance, this tool performs the following steps: (1) it extracts a line range for a function from AST; (2) it uses a lexical scanner to tokenize function's source; (3) for each token it decides whether the token is an operator or an operand, or neither. First of all the token type is used to decide if it is an operator or operand, see Table 4.

If the token type is not enough to distinguish between an operator and an operand; then if tokenize.NAME indicates tokens are Python keywords, they are considered operators; otherwise they are considered operands. McCabe's complexity for functions is calculated directly from AST. Table 5 presents effects of Python statements on cyclomatic complexity score.

3.2.3. Training set generation

Creating a fault predicting tool applicable to many projects can be achieved either by training a universal model, or by training predictors individually for each project [29]. This research adopts the latter approach: for each project it generates a training set using data extracted from the given project repository. Instances in the training set have to be assigned to classes; in this case software functions have to be labelled as either *fault-prone* or *not fault-prone*. In previous studies, such labels were typically assigned by human experts, which is a tedious and expensive process. In order to avoid this step, this tool relies on the following general definition of fault-proneness:

Definition 3.5. For a given revision, function is *fault-prone* if it was fixed in one of K next commits, where the choice of K should depend on the frequency of commits.

The definition of fault proneness can be extended due to the fact that relying on a project architecture warehouse enables mining information in commit logs. For identification of commits as bug-fixing in this research a simple heuristic, frequently used in previous studies, was followed [30, 31].

Definition 3.6. Commit is *bug-fixing* if its log contains any of the following words: *bug*, *fix*, *issue*.

Obviously such a method of generating training data is based on the assumption that bug fixing commits are properly marked and contain only fixes, which is consistent with the best practices for Git [32]. It is worth noting that since this might not be the general case for all projects, the tool in its current format is not recommended for predicting faults in projects that do not follow these practices.

3.2.4. Training and predicting

Training a classifier and making predictions for new instances are the key parts of a fault predictor. For these phases, the tool relies on *GaussianNB* from the *Scikit-learn* (scikit-learn.org) implementation of the Naïve Bayes classifier.

4. Main result

The tool's performance was experimentally assessed on five arbitrarily selected open-source projects of different characteristics: Flask – a web

Table 4. Operator and operand types

OPERATOR_TYPES = [tokenize.OP, tokenize.NEWLINE, tokenize.INDENT, tokenize.DEDENT]

OPERAND_TYPES = [tokenize.STRING, tokenize.NUMBER]

Table 5. Contribution of Python constructs to cyclomatic complexity

Construct	Effect	Reasoning
if	+1	An **if** statement is a single decision
elif	+1	The **elif** statement adds another decision
else	0	Does not cause a new decision - the decision is at the **if**
for	+1	There is a decision at the start of the loop
while	+1	There is a decision at the **while** statement
except	+1	Each **except** branch adds a new conditional path of execution
finally	0	The **finally** block is unconditionally executed
with	+1	The **with** statement roughly corresponds to a **try**/**except** block
assert	+1	The **assert** statement internally roughly equals a conditional statement
comprehension	+1	A *list/set/dict comprehension* of generator expression is equivalent to a **for** loop
lambda	+1	A *lambda function* is a regular function
boolean	+1	Every boolean operator (**and**, **or**) adds a decision point

Table 6. Projects used for evaluation

Project	Location at github.com
Flask	/mitsuhiko/flask
Odoo	/odoo/odoo
GitPython	/gitpython-developers/GitPython
Ansible	/ansible/ansible
Grab	/lorien/grab

Table 7. Summary of projects used for evaluation: projects' revisions (Rv) with corresponding number of commits (Co), branches (Br), releases (Rl) and contributors (Cn)

Project	Rv	Cm	Br	Rl	Cn
Flask	7f38674	2319	16	16	277
Odoo	898cae5	94106	12	79	379
GitPython	7f8d9ca	1258	7	20	67
Ansible	718812d	15935	34	76	1154
Grab	e6477fa	1569	2	0	32

development micro-framework; Odoo – a collection of business apps; GitPython – a library to interact with Git repositories; Ansible – an IT automation system; Grab – a web scraping framework. Analyzed software varies in scope and complexity: from a library with narrow scope, through frameworks, to a powerful IT automation platform and a fully-featured ERP system. All projects are publicly available on GitHub (see Table 6) and are under active development.

Data sets for evaluation were generated from projects using method described in section 3, namely: features were calculated for revision HEAD $\sim$ 100, where HEAD is a revision specified in Table 7; functions were labeled as fault-prone if they were modified in bug-fixing commit between revisions HEAD $\sim$100 and HEAD; data set was truncated to files modified in any commit between revisions HEAD $\sim$100 and HEAD. Table 8 presents total count and incidence of fault-prone functions for each data set.

As defined in section 3, *recall* and *false positive rates* were used to assess the performance of fault predictors. In terms of these metrics, a good fault predictor should achieve: *high recall* – a fault predictor should identify as many faults in the project as possible; if two predictors obtain the same false positive rate, the one with higher recall is preferred, as it will yield more fault-prone functions; *low false positive rate* – code units identified as bug prone require developer action; the predictor with fewer false alarms requires less

Table 8. Data sets used for evaluation

Project	Functions	Fault-prone	% fault-prone
Flask	786	30	3.8
Odoo	1192	50	4.2
GitPython	548	63	11.5
Ansible	752	69	9.2
Grab	417	31	7.4

Table 9. Results for the best predictor

Project	Recall		False positive rate	
	mean	SD	mean	SD
Flask	0.617	0.022	0.336	0.005
Odoo	0.640	< 0.001	0.234	0.003
GitPython	0.467	0.019	0.226	0.003
Ansible	0.522	< 0.001	0.191	0.002
Grab	0.416	0.010	0.175	0.004
In total:	0.531	< 0.03	0.240	< 0.03

human effort, as it returns less functions that are actually not fault-prone.

It is worth noting that Zhang and Zhang [33] argue that a good prediction model should actually achieve both high recall and high precision. However, Menzies et al. [34] advise against using precision for assessing fault predictors, as it is less stable across different data sets than the false positive rate. This study follows this advice.

For this research a stratified 10-fold cross validation was used as a base method for evaluating predicting performance. K-fold cross validation divides instances from the training set into K equal sized buckets, and each bucket is then used as a test set for a classifier trained on the remaining $K-1$ buckets. This method ensures that the classifier is not evaluated on instances it used for learning and that all instances are used for validation.

As bug prone functions were rare in the training sets, folds were stratified, i.e. each fold contained roughly the same proportions of samples for each label.

This procedure was additionally repeated 10 times, each time randomizing the order of examples. This step was added to check whether predicting performance depends on the order of the training set. A similar process was used by other researchers (e.g. [1, 35]).

Main result 1. The fault predictor presented in this research achieved recall up to 0.64 with false positive rate 0.23 (mean recall 0.53 with false positive rate 0.24, see Table 9 for details).

It is worth noting that: the highest recall was achieved for project *Odoo*: 0.640; the lowest recall was achieved for project *Grab*: 0.416; the lowest false positive rate was achieved for project *Flask*: 0.336. For all data sets recall was significantly higher than the false positive rate. The results were stable over consecutive runs; the standard deviation did not exceed 0.03, neither for recall nor for the false positive rate.

Main result 2. This research additionally supports the significance of applying the logarithmic filter, since the fault predictor implemented for this research without using this filter achieved significantly lower mean recall 0.328 with false positive rate 0.108 (see Table 10 for details).

Table 10. Results for the best predictor without the logarithmic filter

Project	Recall		False positive rate	
	mean	SD	mean	SD
Flask	0.290	0.037	0.119	0.004
Odoo	0.426	0.009	0.132	< 0.001
GitPython	0.273	0.016	0.129	0.006
Ansible	0.371	0.012	0.068	< 0.001
Grab	0.219	0.028	0.064	0.005
In total:	0.328		0.108	

It should be emphasised that similar significance was indicated in the case of the detectors for C/C++ and Java projects in [1].

5. Conclusions

In this study, machine learning methods were applied to a software engineering problem of fault prediction. Fault predictors can be useful for directing quality assurance efforts. Prior studies showed that static code features can be used for building practical fault predictors for C/C++ and Java projects. This research demonstrates that these techniques also work for Python, a pop-

ular programming language that was omitted in previous research. The tool resulting from this research is a function-level fault prediction tool for Python projects. Its performance was experimentally assessed on five open-source projects. On selected projects the tool achieved recall up to 0.64 with false positive rate 0.23, mean recall 0.53 with false positive rate 0.24. Leading fault predictors trained on NASA data sets achieved higher mean recall 0.71 with similar false positive rate 0.25 [1]. Labour intensive, manual code inspections can find about 60% of defects [36]. This research is close to reaching a similar level of recall. The performance of this tool can be perceived as satisfactory, certainly proving the hypothesis that predicting faults for Python programs has a similar potential to that of C/C++ and Java programs, and that more thorough future research in this area is worth conducting.

5.1. Threats to validity

Internal There are no significant threats to internal validity. The goal was to take an approach inspired by the experiments conducted by Menzies et al. [1] The experimental results for Python demonstrated to be consistent with the ones reported for C/C++ and Java, claiming that: static code features are useful for the identification of faults, fault predictors using the Naïve Bayes classifier perform well, however, using a logarithmic filter is encouraged, as it improves predicting performance. Using other methods of extracting features used for machine learning (i.e. Python features which are absent in C/C++ or Java), could potentially lead to a better performance of the tool.

External There are threats to external validity. The results obtained in this research are not valid for generalization from the context in which this experiment was conducted to a wider context. More precisely, the range of five arbitrarily selected software projects provides experimental evidence that this direction of research is worth pursuing; however, by itself it does not provide enough evidence for general conclusions and more thorough future research is required. Also the tool performance was assessed only in terms of recall and false positive rates, it has not been actually verified in practice. It is thus possible that the tool current predicting ability might prove not good enough for practical purposes and its further development will be required. Therefore, the conclusion of the universal practical applicability of such an approach cannot be drawn yet.

Construct There are no significant threats to construct validity. In this approach the authors were not interested in deciding whether it is a well selected machine learning technique, project attributes used for learning or the completeness of fault proneness definition for the training-set that were mainly contributing to the tool performance. The important conclusion was that the results obtained do not exclude but support the hypothesis, that automated fault prediction in Python allows to obtain accuracy comparable to the results obtained for other languages and to human-performed fault prediction, hence they encourage more research in this area. Thus, the results provided in this paper serve as an example and the rough estimation of predicting performance expected nowadays from fault predictors using static code features. There are few additional construct conditions worth mentioning. As discussed in section 3, the tool training set generation method relies on project change logs being part of the project architecture warehouse. If bug-fixing commits are not properly labelled, or contain not only fixes, then the generated data sets might be skewed. Clearly, the performance of the tool can be further improved, as it is not yet as good as the performance of fault predictors for C/C++ and Java; the current result is a good start for this improvement. Comparing the performance of classifiers using different data sets is not recommended, as predictors performing well on one set of data might fail on another.

Conclusion There are no significant threats to conclusion validity. Fault recall (detection rate) alone is not enough to properly assess the performance of a fault predictor (i.e. a trivial fault predictor that labels all functions as fault-prone achieves total recall), hence the focus on both recall (detection) and false positives (false alarms).

Obviously the false positive rate of a fault predictor should be lower than its recall, as a predictor randomly labelling p of functions as fault-prone on average achieves a recall and false positive rate of p. This has been achieved in this study, similarly to [1]. From the practical perspective, in this research the goal recognizing automatically as many relevant (erroneous) functions as possible, which later should be revised manually by programmers; that is the authors were interested in achieving high recall and trading precision for recall if needed. From the perspective of this research goals, evaluating classifiers by measures other than those used in [1] (i.e. using other elements in the confusion matrix) was not directly relevant for the conclusions presented in this paper.

5.2. Future research

Additional features As mentioned in section 3, static code metrics are only a subset of features that can be used for training fault predictors. In particular, methods utilizing previous defect data, such as, [37] can also be useful for focusing code inspection efforts [38, 39]. Change data, such as code churn or fine-grained code changes were also reported to be significant bug indicators [40–42]. Adding support for these features might augment their fault predicting capabilities. Moreover, further static code features, such as object oriented metrics defined by Chidamber and Kemerer [43] can be used for bug prediction [32, 44]. With more attributes, adding a feature selection step to the tool might also be beneficial. Feature selection can also improve training times, simplify the model and reduce overfitting.

Additional algorithms The tool uses a Naïve Bayes classifier for predicting software defects. In preliminary experiments different learning algorithms were assessed, but they performed significantly worse. It is possible that with more features supplied and fine-tuned parameters these algorithms could eventually outperform the Naïve Bayes classifier. Prediction efficiency could also be improved by including some strategies for eliminating class imbalance [45] in the data sets. Researchers also keep proposing more sophisticated methods for identifying bug-fixing commits than the simple heuristic used in this research, in particular high-recall automatic algorithms for recovering links between bugs and commits have been developed. Integrating algorithms, such as [46] into a training set generation process could improve the quality of the data and, presumably, tool predicting performance.

Additional projects In preliminary experiments, a very limited number of Python projects were used for training and testing. Extending the set of Python projects contributing to the training and testing sets is needed to generalize the conclusions. The selection of additional projects should be conducted in a systematic manner. A live code could be used for predictor evaluation [22], which means introducing predictors into the development toolsets used by software developers in live software projects. The next research steps should involve a more in-depth discussion about the findings on the Python projects, in particular identification why in some projects the proposed techniques have a better performance than in other projects.

References

[1] T. Menzies, J. Greenwald, and A. Frank, "Data mining static code attributes to learn defect predictors," *IEEE Transactions on Software Engineering*, Vol. 33, No. 1, 2007, pp. 2–13.

[2] E.W. Dijkstra, "Letters to the editor: go to statement considered harmful," *Communications of the ACM*, Vol. 11, No. 3, 1968, pp. 147–148.

[3] J. McCarthy, P.W. Abrams, D.J. Edwards, T.P. Hart, and M.I. Levin, *Lisp 1.5 programmer's manual*. The MIT Press, 1962.

[4] W. Royce, "Managing the development of large software systems: Concepts and techniques," in *Technical Papers of Western Electronic Show and Convention (WesCon)*, 1970, pp. 328–338.

[5] K. Beck, "Embracing change with extreme programming," *IEEE Computer*, Vol. 32, No. 10, 1999, pp. 70–77.

[6] R. Kaufmann and D. Janzen, "Implications of test-driven development: A pilot study," in *Companion of the 18th annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications*, ser.

OOPSLA '03. New York, NY, USA: ACM, 2003, pp. 298–299.

[7] R. Dąbrowski, "On architecture warehouses and software intelligence," in *Future Generation Information Technology*, ser. Lecture Notes in Computer Science, T.H. Kim, Y.H. Lee, and W.C. Fang, Eds., Vol. 7709. Springer, 2012, pp. 251–262.

[8] R. Dąbrowski, K. Stencel, and G. Timoszuk, "Software is a directed multigraph," in *5th European Conference on Software Architecture ECSA*, ser. Lecture Notes in Computer Science, I. Crnkovic, V. Gruhn, and M. Book, Eds., Vol. 6903. Essen, Germany: Springer, 2011, pp. 360–369.

[9] R. Dąbrowski, G. Timoszuk, and K. Stencel, "One graph to rule them all (software measurement and management)," *Fundamenta Informaticae*, Vol. 128, No. 1-2, 2013, pp. 47–63.

[10] G. Tassey, "The economic impacts of inadequate infrastructure for software testing," National Institute of Standards and Technology, Tech. Rep., 2002. [Online]. https://pdfs.semanticscholar.org/9b68/5f84da00514397d9af7f27cc0b7db7df05c3.pdf

[11] T. Menzies, Z. Milton, B. Turhan, B. Cukic, Y. Jiang, and A.B. Bener, "Defect prediction from static code features: Current results, limitations, new approaches," *Automated Software Engineering*, Vol. 17, No. 4, 2010, pp. 375–407.

[12] S. Lessmann, B. Baesens, C. Mues, and S. Pietsch, "Benchmarking classification models for software defect prediction: A proposed framework and novel findings," *IEEE Transactions on Software Engineering*, Vol. 34, No. 4, 2008, pp. 485–496.

[13] T. Hall, S. Beecham, D. Bowes, D. Gray, and S. Counsell, "A systematic literature review on fault prediction performance in software engineering," *IEEE Transactions on Software Engineering*, Vol. 38, No. 6, 2012, pp. 1276–1304.

[14] B.W. Boehm, "Software risk management," in *ESEC '89, 2nd European Software Engineering Conference*, ser. Lecture Notes in Computer Science, C. Ghezzi and J.A. McDermid, Eds., Vol. 387. Springer, 1989, pp. 1–19.

[15] T.M. Khoshgoftaar and N. Seliya, "Fault prediction modeling for software quality estimation: Comparing commonly used techniques," *Empirical Software Engineering*, Vol. 8, No. 3, 2003, pp. 255–283.

[16] A.A. Porter and R.W. Selby, "Empirically guided software development using metric-based classification trees," *IEEE Software*, Vol. 7, No. 2, 1990, pp. 46–54.

[17] T. Menzies, J. DiStefano, A. Orrego, and R.M. Chapman, "Assessing predictors of software defects," in *Proceedings of Workshop Predictive Software Models*, 2004.

[18] T. Menzies, R. Krishna, and D. Pryor, *The Promise Repository of Empirical Software Engineering Data*, North Carolina State University, Department of Computer Science, (2015). [Online]. http://openscience.us/repo

[19] T. Menzies, J.S.D. Stefano, K. Ammar, K. McGill, P. Callis, R.M. Chapman, and J. Davis, "When can we test less?" in *9th IEEE International Software Metrics Symposium (METRICS)*, Sydney, Australia, 2003, p. 98.

[20] T. Menzies, J.S.D. Stefano, and M. Chapman, "Learning early lifecycle IV&V quality indicators," in *9th IEEE International Software Metrics Symposium (METRICS)*, Sydney, Australia, 2003, pp. 88–97.

[21] T. Menzies and J.S.D. Stefano, "How good is your blind spot sampling policy?" in *8th IEEE International Symposium on High-Assurance Systems Engineering (HASE)*. Tampa, FL, USA: IEEE Computer Society, 2004, pp. 129–138.

[22] M. Lanza, A. Mocci, and L. Ponzanelli, "The tragedy of defect prediction, prince of empirical software engineering research," *IEEE Software*, Vol. 33, No. 6, 2016, pp. 102–105.

[23] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*, ser. Adaptive computation and machine learning. The MIT Press, 2012.

[24] R.C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall, 2008.

[25] T.J. McCabe, "A complexity measure," *IEEE Transactions on Software Engineering*, Vol. 2, No. 4, 1976, pp. 308–320.

[26] M.H. Halstead, *Elements of Software Science (Operating and Programming Systems Series)*. New York, NY, USA: Elsevier Science Ltd, 1977.

[27] N.E. Fenton and S.L. Pfleeger, *Software Metrics: A Rigorous and Practical Approach*, 2nd ed. Boston, MA, USA: Course Technology, 1998.

[28] C. Ebert and J. Cain, "Cyclomatic complexity," *IEEE Software*, Vol. 33, No. 6, 2016, pp. 27–29.

[29] F. Zhang, A. Mockus, I. Keivanloo, and Y. Zou, "Towards building a universal defect prediction model," in *11th Working Conference on Mining Software Repositories, MSR*, P.T. Devanbu, S. Kim, and M. Pinzger, Eds. Hyderabad, India: ACM, 2014, pp. 182–191.

[30] S. Kim, "Adaptive bug prediction by analyzing project history," Ph.D. dissertation, University of California at Santa Cruz, Santa Cruz, CA, USA, 2006, aAI3229992.

[31] S. Kim, T. Zimmermann, K. Pan, and E.J. Whitehead, Jr., "Automatic identification of bug-introducing changes," in *21st IEEE/ACM International Conference on Automated Software Engineering (ASE)*. Tokyo, Japan: IEEE Computer Society, 2006, pp. 81–90.

[32] T. Gyimóthy, R. Ferenc, and I. Siket, "Empirical validation of object-oriented metrics on open source software for fault prediction," *IEEE Transactions on Software Engineering*, Vol. 31, No. 10, 2005, pp. 897–910.

[33] H. Zhang and X. Zhang, "Comments on 'Data mining static code attributes to learn defect predictors'," *IEEE Transactions on Software Engineering*, Vol. 33, No. 9, 2007, pp. 635–637.

[34] T. Menzies, A. Dekhtyar, J.S.D. Stefano, and J. Greenwald, "Problems with precision: A response to 'Comments on data mining static code attributes to learn defect predictors'," *IEEE Transactions on Software Engineering*, Vol. 33, No. 9, 2007, pp. 637–640.

[35] M.A. Hall and G. Holmes, "Benchmarking attribute selection techniques for discrete class data mining," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 6, 2003, pp. 1437–1447.

[36] F. Shull, V.R. Basili, B.W. Boehm, A.W. Brown, P. Costa, M. Lindvall, D. Port, I. Rus, R. Tesoriero, and M.V. Zelkowitz, "What we have learned about fighting defects," in *8th IEEE International Software Metrics Symposium (METRICS)*. Ottawa, Canada: IEEE Computer Society, 2002, p. 249.

[37] S. Kim, T. Zimmermann, E.J. Whitehead, Jr., and A. Zeller, "Predicting faults from cached history," in *29th International Conference on Software Engineering (ICSE 2007)*. Minneapolis, MN, USA: IEEE, 2007, pp. 489–498.

[38] F. Rahman, D. Posnett, A. Hindle, E.T. Barr, and P.T. Devanbu, "BugCache for inspections: hit or miss?" in *SIGSOFT/FSE'11 19th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-19) and ESEC'11: 13rd European Software Engineering Conference (ESEC-13)*, T. Gyimóthy and A. Zeller, Eds. Szeged, Hungary: ACM, 2011, pp. 322–331.

[39] C. Lewis, Z. Lin, C. Sadowski, X. Zhu, R. Ou, and E.J.W. Jr., "Does bug prediction support human developers? Findings from a Google case study," in *35th International Conference on Software Engineering, ICSE*, D. Notkin, B.H.C. Cheng, and K. Pohl, Eds. San Francisco, CA, USA: IEEE / ACM, 2013, pp. 372–381.

[40] N. Nagappan and T. Ball, "Use of relative code churn measures to predict system defect density," in *27th International Conference on Software Engineering (ICSE)*, G. Roman, W.G. Griswold, and B. Nuseibeh, Eds. ACM, 2005, pp. 284–292.

[41] A.E. Hassan, "Predicting faults using the complexity of code changes," in *31st International Conference on Software Engineering, ICSE*. Vancouver, Canada: IEEE, 2009, pp. 78–88.

[42] E. Giger, M. Pinzger, and H.C. Gall, "Comparing fine-grained source code changes and code churn for bug prediction," in *Proceedings of the 8th International Working Conference on Mining Software Repositories, MSR*, A. van Deursen, T. Xie, and T. Zimmermann, Eds. ACM, 2011, pp. 83–92.

[43] S.R. Chidamber and C.F. Kemerer, "Towards a metrics suite for object oriented design," in *Sixth Annual Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '91)*, A. Paepcke, Ed. Phoenix, Arizona, USA: ACM, 1991, pp. 197–211.

[44] R. Shatnawi, "Empirical study of fault prediction for open-source systems using the Chidamber and Kemerer metrics," *IET Software*, Vol. 8, No. 3, 2014, pp. 113–119.

[45] N.V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: Special issue on learning from imbalanced data sets," *ACM SIGKDD Explorations Newsletter*, Vol. 6, No. 1, Jun. 2004, pp. 1–6.

[46] R. Wu, H. Zhang, S. Kim, and S. Cheung, "ReLink: recovering links between bugs and changes," in *SIGSOFT/FSE'11 19th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-19) and ESEC'11: 13rd European Software Engineering Conference (ESEC-13)*, T. Gyimóthy and A. Zeller, Eds. Szeged, Hungary: ACM, 2011, pp. 15–25.

[47] T. Gyimóthy and A. Zeller, Eds., *SIGSOFT-/FSE 11 Proceedings of the 19th ACM SIGSOFT Symposium on Foundations of Software Engineering*. Szeged, Hungary: ACM, 2011.

Appendix

The implementation of the method used in this study for predictor evaluation is outlined below, it can be used to reproduce results of the experiments.

```
# imports available on github.com
import git
import numpy as np
from sklearn import cross_validation
from sklearn import metrics
from sklearn import naive_bayes
from sklearn import utils
from scary import dataset
from scary import evaluation

def run():
        projects = [
                "path/to/flask",
                "path/to/odoo",
                "path/to/GitPython",
                "path/to/ansible",
                "path/to/grab",
        ]
        classifier = naive_bayes.GaussianNB()
        EvaluationRunner(projects, classifier).evaluate()

class EvaluationRunner:
        def __init__(self, projects, classifier, from_revision="HEAD~100", to_revision="HEAD",
                shuffle_times=10, folds=10):
                self.projects = projects
                self.classifier = classifier
                self.from_revision = from_revision
                self.to_revision = to_revision
                self.shuffle_times = shuffle_times
                self.folds = folds

        def evaluate(self):
                total_score_manager = self.total_score_manager()
                for project in self.projects:
                        project_score_manager = self.project_score_manager()
                        training_set = self.build_training_set(project)
                        for data, target in self.shuffled_training_sets(training_set):
                                predictions = self.cross_predict(data, target)
                                confusion_matrix = self.confusion_matrix(predictions, target)
                                total_score_manager.update(confusion_matrix)
                                project_score_manager.update(confusion_matrix)
                        self.report_score(project, project_score_manager)
                self.report_score("TOTAL", total_score_manager)

        def project_score_manager(self):
                return ScoreManager.project_score_manager()

        def total_score_manager(self):
                return ScoreManager.total_score_manager()
```

```
50
51          def build_training_set(self, project):
52                  repository = git.Repo(project)
53                  return dataset.TrainingSetBuilder.build_training_set(repository,
54                          self.from_revision, self.to_revision)
55
56          def shuffled_training_sets(self, training_set):
57                  for _ in range (self.shuffle_times):
58                          yield utils.shuffle(training_set.features, training_set.classes)
59
60          def cross_predict(self, data, target):
61                  return cross_validation.cross_val_predict(self.classifier, data, target,
62                          cv=self.folds)
63
64          def confusion_matrix(self, predictions, target):
65                  confusion_matrix = metrics.confusion_matrix(target, predictions)
66                  return evaluation.ConfusionMatrix(confusion_matrix)
67
68          def report_score(self, description, score_manager):
69                  print(description)
70                  score_manager.report()
71
72  class ScoreManager:
73          def __init__(self, counters):
74                  self.counters = counters
75
76          def update (self, confusion_matrix):
77                  for counter in self.counters:
78                          counter.update(confusion_matrix)
79
80          def report(self):
81                  for counter in self.counters:
82                          print(counter.description, counter.score)
83
84          @classmethod
85          def project_score_manager(cls):
86                  counters = [MeanScoreCounter(RecallCounter),
87                          MeanScoreCounter(FalsePositiveRateCounter),]
88                  return cls(counters)
89
90          @classmethod
91          def total_score_manager(cls):
92                  counters = [RecallCounter(),
93                          FalsePositiveRateCounter(),]
94                  return cls(counters)
95
96  class BaseScoreCounter:
97          def update (self, confusion_matrix):
98                  raise NotImplementedError
99
100         @property
101         def score (self):
102                 raise NotImplementedError
103
```

```
104         @property
105         def decription(self):
106                 raise NotImplementedError
107
108 class MeanScoreCounter(BaseScoreCounter):
109         def __init__(self, partial_counter_class):
110                 self.partial_counter_class= partial_counter_class
111                 self.partial_scores = []
112
113         def update(self, confusion_matrix):
114                 partial_score = self.partial_score(confusion_matrix)
115                 self.partial_scores.append(partial_score)
116
117         def partial_score(self, confusion_matrix):
118                 partial_counter = self.partial_counter_class()
119                 partial_counter.update(confusion_matrix)
120                 return partial_counter.score
121
122         @property
123         def score (self):
124                 return np.mean(self.partial_scores), np.std(self.partial_scores)
125
126         @property
127         def description(self):
128                 return "mean {}".format(self.partial_counter_class().description)
129
130 class RecallCounter(BaseScoreCounter):
131         def __init__(self):
132                 self.true_positives = 0
133                 self.false_negatives = 0
134
135         def update(self, confusion_matrix):
136                 self.true_positives += confusion_matrix.true_positives
137                 self.false_negatives += confusion_matrix.false_negatives
138
139         @property
140         def score(self):
141                 return self.true_positives/(self.true_positives+self.false_negatives)
142
143         @property
144         def description(self):
145                 return "recall"
146
147 class FalsePositiveRateCounter(BaseScoreCounter):
148         def __init__(self):
149                 self.false_positives = 0
150                 self.true_negatives = 0
151
152         def update (self, confusion_matrix):
153                 self.false_positives += confusion_matrix.false_positives
154                 self.true_negatives += confusion_matrix.true_negatives
155
156         @property
157         def score (self):
```

```
158                 return self.false_positives/(self.false_positives+self.true_negatives)
159
160         @property
161         def description(self):
162                 return "false positive rate"
163
164 if __name__ == "__main__":
165         run ()
```

e-Informatica Software Engineering Journal, Volume 12, Issue 1, 2018, pages: 217–236, DOI 10.5277/e-Inf180109

Milestone-Oriented Usage of Key Performance Indicators – An Industrial Case Study

Miroslaw Staron*, Kent Niesel**, Niclas Bauman**
* *Computer Science and Engineering, Chalmers / University of Gothenburg*
** *Volvo Car Group*
miroslaw.staron@gu.se, kent.niesel@volvocars.com, niclas.bauman@volvocars.com

Abstract

Background: Key Performance Indicators are a common way of quantitative monitoring of project progress in modern companies. Although they are widely used in practice, there is little evidence on how they are set, and how many of them are used in large product development projects.
Goal: The goal of this paper is to explore how KPIs are used in practice in a large company. In particular, it is explored whether KPIs are used continuously or only during short, predefined periods of time. It is also explored whether software-related KPIs are reported differently from non-software-related KPIs.
Method: A case study of 12 projects at the Volvo Car Group in Sweden was conducted. The data from the project progress reporting on tools was collected and triangulated with data from interviews conducted with experts from the company.
Results: KPIs are reported mostly before the milestones and the manual assessment of their status is equally important as the automated data provision in the KPI reporting system. The trend of reporting software-related KPIs is very similar to the non-software-related KPIs.
Conclusions: Despite the documented good practices of using KPIs for project monitoring, it is difficult to develop a clear status-picture solely using quantitative data from progress reporting tools. It was also shown that the trends in reporting the software-related KPIs are similar to the trends in reporting the non-software related KPIs.

Keywords: software metrics, key performance indicator, project management, case study

1. Introduction

Monitoring large product development projects is a challenging task for project management teams. Project managers, together with sub-project managers at different levels, quality managers and line managers, often use quantitative data to present the progress of projects and the readiness of their products [1]. Key Performance Indicators (KPIs) are used for the purpose of monitoring progress, to capture quantitative data and interpret it [2]. Although this practice is very common and well-known, there are not many studies on how this reporting is done in practice, e.g. how often the KPIs are reported, how many KPIs are used and how quantitative data is used in setting the values of the KPIs. In the literature, the classical use of KPIs is to continuously report and monitor the progress of the development, which usually leads to using KPIs for decision support and dissemination of information about the project status [3]. This study aims at analysing how the literature evidence is aligned with the practice of using KPIs in a large company.

Embedded software development projects require synchronization between software-related and non-software-related sub-projects in order to result in a complete product. Project managers, however, often seek advice on whether the projects should follow a more software-inspired agile way of planning (and thus adjust to the software project management practices) or follow

more strict, hardware-inspired project planning and monitoring (thus putting dedicated reporting requirements on software projects). In this paper, the authors explore how large software projects are managed and whether the usage of KPIs is more software agile-like or hardware upfront planning-like.

1.1. Problem statement

The paper explores the problem of understanding the practice of using KPIs. In theory, KPIs should be reported and monitored continuously, in order to provide an up-to-date status of the project. However, in practice, this continuous reporting requires resources and scales poorly with the number of KPIs. It is also the case that, if the number of KPIs grows, the probability increases that multiple KPIs monitor the same or related issues.

The contribution of this paper is analysing both KPIs for software-related KPIs and for the entire project. The state-of-the-art in this area considers either only software development projects, or focuses on project-management aspects. Therefore, it is important to study the KPIs in embedded software projects, where the software development sub-project is contrasted with non-software sub-projects.

1.2. Research objectives and questions

The general research methodology applied in this work is a case study; the methodology which emphasizes close collaboration between industry and academia, and results in changes in hosting organizations. To begin with the following research question is addressed – How are KPIs for monitoring project progress used in practice in a large product development organization? Steyn and Stoker [4] recognized this as an issue, and provided the evidence that different ways of using KPIs impact the performance of development projects. The use of KPIs can also determine whether the company uses the traditional approach to performance monitoring, or it uses the modern principles of Neely et al. [5]. In the research twelve projects were studied focusing on such aspects of reporting as:

- How are KPIs defined? – to understand the structure of KPIs used in industry.
- How often are KPIs reported in practice? – to explore the frequency and thus the cost of reporting KPIs, and to understand how timely the KPI information is provided.
- Who is responsible for reporting and acting upon the definitions of KPIs? – to understand the stakeholders in the process of KPI reporting and decision making.
- How can we statistically identify dependencies between KPIs? – to explore whether KPIs are independent from one another, and therefore to understand whether the number of KPIs is sufficient or too extensive.
- Is there a difference between software-related KPIs and non-software-related KPIs? – to explore whether the software development KPIs are reported differently than the non-software development ones.

1.3. Context

This work studies a large product development organization – the Volvo Car Group, a Swedish vehicle manufacturer. The analysis encompassed 12 car development projects where the number of KPIs varies from 252 to 552 per project. The following definition of a KPI was used – KPI *(Key Performance Indicator)* is a customizable business metric utilized to visualize the status and trends in an organization. A KPI has an owner (a stakeholder according to ISO/IEC 15939 [6]), an interpretation (an analysis model according to ISO/IEC 15939) and is linked to a business strategy of the organization. This definition of a KPI is consistent with the use of the term in well-established methodologies, such as the Balance Scorecard [7].

The remaining of the paper is structured as follows. Section 2 presents the theoretical framing of work. Section 3 describes the design of the case study. Section 4 presents the results and answers to the research questions. Section 5 discusses the results in the light of the existing body of

knowledge. Section 6 summarizes the paper and presents conclusions.

2. Background

In order to study the use of KPIs in the company the authors used a set of models showing how the reporting process is done. The models are presented in Figure 1. The models are divided into four groups of activities:

- storage: the way in which information, needed to calculate the KPI, is stored – it could be either a database, such as a product article database or personal assessment of, e.g., whether the quality of a requirement is sufficient,
- extraction: the way in which the information is provided to KPI systems – it could be manual reporting or the automatic extraction of information using a script (for example by counting the number of defects reported in a database),
- analysis: the set of methods for analysing the values of KPI, they assess the status (set the colour of a KPI – green, yellow or red) – it could be an algorithm, using a set of pre-defined criteria, or a manual assessment,
- presentation: grouping activities related to the presentation of the material, which can also be either manual or automated – the manual presentation can be in the form of an MS PowerPoint presentation and the automated one can be a web-based dashboard with indicators [8].

These four groups of activities are based on the measurement information model defined in the ISO/IEC 15939 standard (Software and Systems Engineering – Measurement Processes), [6].

The set of models used for the theoretical framing of the KPI usage, comprises four models which have distinct characteristics.

The most basic model is manual reporting, the analysis and presentation of the KPI values, as initially presented by Kapplan and Norton, as part of the Balanced Scorecard methodology [9]. In this model (M), the focus of KPI usage is on the periodical reporting and monitoring of organizational performance. The data to calculate the values of KPIs is often available through individuals (e.g. by filling in reports) and needs manual assessment. The KPIs, in the M model, are often updated periodically and are prone to missing data points, however, it is very flexible. This model can be observed in the studies discussion early adoption of the Balanced Scorecard [9].

The next model, which is more advanced, is the M-A model, where the extraction activities are automated, but the assessment of the KPIs status is manual. This kind of model is prescribed by many project management tools and methodologies which focus on the quantitative assessment of project progress and performance. An example of such a method is PRINCE2 [10]. The M-A model can be observed in modern companies utilizing business intelligence tools. The KPIs in the M-A model are updated continuously and are analysed periodically; in practice this means the same disadvantages as the M model with a reduction of problems coming from the missing data points. The M-A model can be exemplified by such cases as surveys for customer satisfaction [11].

The next model is the A-M model, where most of the data extraction and presentation tasks are automated. The assessment of the values is, however, still manual. An example of such an assessment is the quality of the product under development by counting the number of defects discovered during testing. As the automation of the extraction and presentation is used, the KPIs in this model are often used as measures, and visualized as trends – since they are collected continuously. However, their assessment is periodical, which means that the status is available at certain points of time. Thus KPIs in this model are more difficult to interpret, but easier to visualize [12]. An example of this kind of a model is the set of KPIs at Volvo Car Group where the data collection is automated but the setting of colour is manual [13].

Finally, the most advanced model is the A model, where all tasks are automated. The stakeholders of the KPI pre-define rules for analysis and these rules are applied automatically for the measures collected. This kind of model has been shown

to be an efficient way of collecting the information and supporting decisions [14,15], [16]. KPIs which are used in this way require maintenance (evolving criteria, updating data extraction programs), but require no manual effort on a daily basis. It is the automation that makes it very attractive for modern companies. The main disadvantage of this model is, however, the fact that not everything can be calculated automatically, which in practice leads to the degradation of the A-M or M-A models. The main advantage, on the other hand, is the ability to provide status assessment continuously; thus enabling the development of information radiators or dashboards spreading information across the project team. Examples of this type of reporting can be found at such companies as Ericsson.

The set of models used in this study as the theoretical framework, allowed to clearly identify the patterns KPI use. The way of updating KPIs in these models allowed also to identify the possibility of visualizing the status in the long run – potential for the development of dashboards.

3. Case study design

This section describes the design of the study, following the guidelines by Runeson et al. [17].

3.1. Research questions

The following research question were addressed – *How are KPIs for monitoring project progress used in practice in a large product development organization?* In order to address the question, a number of organizations within the Volvo Car Group were studied – they were involved in product development, manufacturing engineering, provisioning of parts for production and contract management. The study encompassed twelve projects and focused on such aspects of reporting as:

- How are KPIs defined? – to understand the structure of KPIs used in industry.
- How often are KPIs reported in practice? – to explore the frequency and thus the cost of reporting KPIs, and to understand how timely the KPI information is provided.
- Who is responsible for reporting and acting upon the definitions of KPIs? – to understand the stakeholders in the process of KPI reporting and decision making.
- How can we statistically identify dependencies between KPIs? – to explore whether KPIs are independent from one another, and therefore to understand whether the number of KPIs is sufficient or too extensive.
- Is there a difference between software-related KPIs and non-software-related KPIs? – to explore whether the software development KPIs are reported differently than the non-software development ones.

Exploring these questions, provides a possibility to understand whether there is a minimum viable set of KPIs to be used in a project, and how to construct a dashboard for visualizing the status of the development in an automated way. For example, in order to construct a real-time dashboard as advocated by Azvine et al. in the context of telecommunication industry [18]. Understanding whether the status of a KPI (or its colour) is usually set using quantitative data from source systems, provides us with a possibility to automate the process of setting the KPI status and thus decrease the cost of project monitoring without a decrease in its quality.

Understanding how KPIs are used in practice requires a combination of analyses of data from different sources and, therefore, two different sources of data collection were triangulated – documents at the company (in the form of a project status reporting tool) and interviews with stakeholders who report on the progress.

3.2. Case and subject selection

The study presented in this paper was conducted over a period of six months. The research was done based on interviews with stakeholders at multiple units of the company – Electrical System and Electrical Propulsion, Powertrain, Chassis, Purchasing and Manufacturing Engineering. The interviews are complemented with the statistical analyses of historical KPI change data from finished and ongoing projects. The statistical analyses are done based on constructing

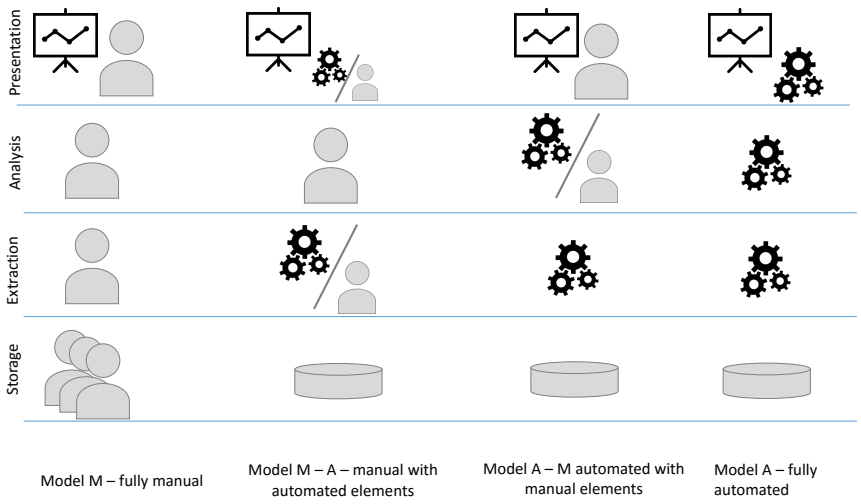


Figure 1. KPI reporting models

co-dependencies, co-change models of KPIs for the selected set of 12 projects.

The projects are selected based on their characteristics – from minor year-model update projects to large complete new vehicle platform projects. The aim was to cover a variety of projects. Table 1 summarizes the characteristics of each project.

The minor model update projects are usually limited to the updates of vehicle parts in this study called disciplines[1] (e.g. new infotainment software, engine control software update), and therefore are limited in scope, but not in the process as all process steps need to be conducted. Studying the smaller projects helps to establish the minimum number of KPIs , whereas studying the large projects allowed to understand how many KPIs are required in large projects. In the table, the number of KPIs as a proxy for the size of the project is used, because in these analyses defined the number of KPIs to correlate with such project parameters as project length, effort and cost.

The number of KPIs reflects the size of the project as larger projects tend to require more monitoring and control than smaller ones. Therefore, project B is the largest one, both in terms of the number of KPIs and the size (confirmed through interviews). Project A is in the middle of the size spectrum (386 KPIs in the set of 252–552 KPIs) and was chosen as a good pilot project to study in detail; the data from this project was used as examples in Section 4. The industrial partners in the research provided this information and also showed evidence for that. However, these numbers cannot be reported outside the company.

For the interviews, the subjects were selected based on their experience. All subjects had had over 10 years of experience in project management and also in managing software and car development projects at the studied company. Each of the subjects was recommended by his or her manager as the most knowledgeable person in that area. All subjects were involved in the subset of the projects studied, although not all respondents were involved in all projects.

3.3. Data collection procedures

This research study was divided into four distinct parts, as presented in Figure 2 – statistical analysis of the co-changes of KPIs and interviews about using KPIs and future needs.

Figure 3 presents an overview of the research process for the statistical KPI co-change analysis. The process comprises four steps – starting from the exporting of the KPI change data from the database and finishing with the prioritization of

[1]Disciplines are the parts of development, e.g. active safety systems development, engine development, powertrain electronic control unit development, new production line development.

Table 1. Main characteristics of the projects studied

Project	KPIs	Characteristics
Project A (pilot)	386	Minor year model update of a mature car model. Development done on a single site, includes all disciplines.
Project B	552	New platform development project. Development done on a single site, includes all disciplines.
Project C	396	New functionality development for an existing platform. Single site, including a subset of disciplines.
Project D	382	New functionality development project. Development done on a single site, includes few disciplines only.
Project E	257	New functionality development project. Development done on a single site, includes few disciplines only.
Project F	442	New engine development project. Development done on a single site, includes few disciplines only.
Project G	305	New functionality development project. Development done on a single site, includes a subset of disciplines only.
Project H	252	New functionality development project. Development done on a single site, includes few disciplines only.
Project I	421	New functionality development project. Development done on a single site, includes few disciplines only.
Project J	342	New functionality development project. Development done on multiple sites, includes all disciplines.
Project K	494	New functionality development project. Development done on a single site (different than projects A–J), includes few disciplines only.
Project L	431	New engine development project. Similar to project F, but on a different engine. Development done on a single site, includes few disciplines only.

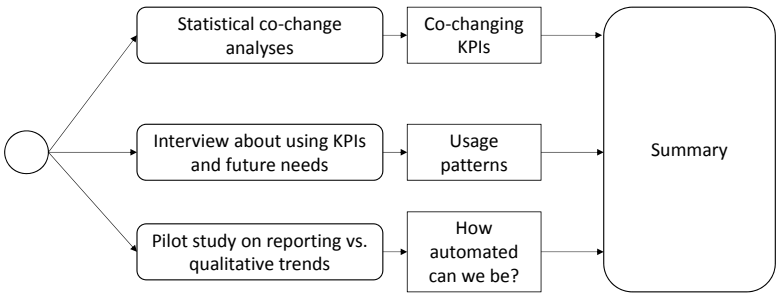


Figure 2. Overview of the research design

candidate KPIs to remove. As shown in the figure, the export of the data from the database in step 1 results in one KPI change per project, and the analyses in step 1 and step 2 are conducted per project. The analyses in step 3 and step 4 are done for all projects, by consolidating the results from each individual project.

In step 1, the exports result in text files (.csv) with the change analyses in the format: <kpi name, value, change date>. Grouping these changes results in the statistics of how often each KPI changed together with another KPI.

The interviews were conducted with 12 different respondents – project managers (1 person), sub-project managers (6), unit project managers (4) and quality responsible (1 person). Each of these people represented a different department at the Volvo Car Group, and each worked with a number of different projects (including the set of projects in our sample). They represented departments responsible for powertrain development, purchasing, quality management, electrical system development (including software) and interior development. The following questions were asked:

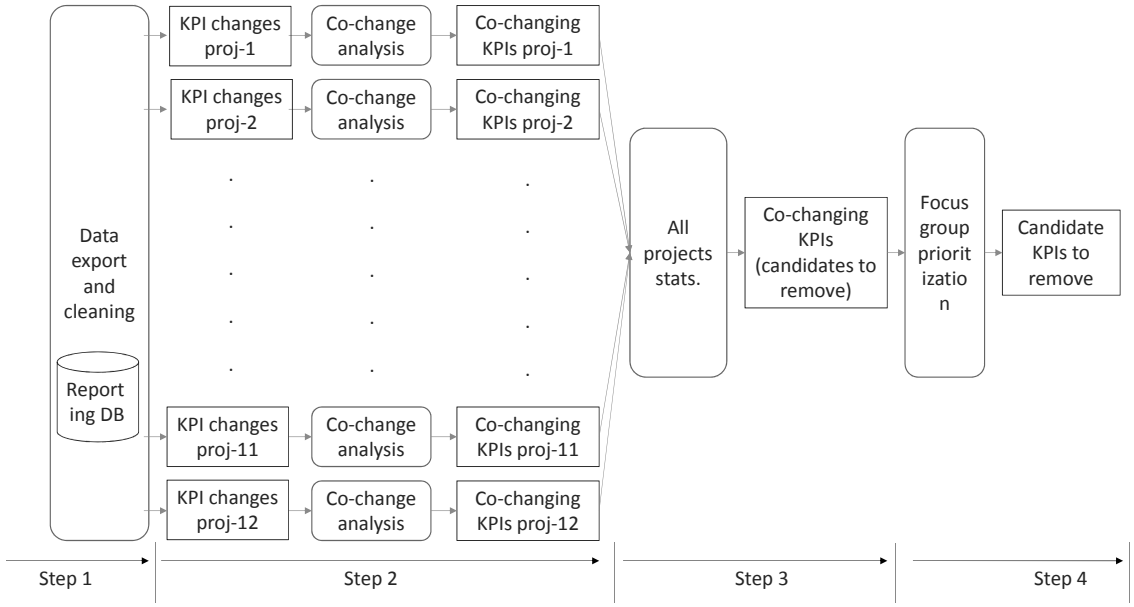


Figure 3. Overview of the statistical analyses

– Which elements of the construction are you responsible for?
– What is the main focus of your work with KPIs?
– How do you update KPIs today? – daily, before the milestone or another frequency?
– When do you discuss the status of the indicators and with whom?
– Which of your indicators are "automatically" imported from other systems?

For each question, the answers were discussed. We also asked about the dependencies between KPIs, their definitions, asked for an exemplification of how the stakeholders work with the persons who define the KPIs.

3.4. Analysis procedures

The first analysis of the patterns of changing KPIs was done by visualizing the changes using a heatmap [19], [20], which is a graphical representation of contingency tables. The heatmaps allow us to:

– identify KPIs which frequently change in the project,
– identify which KPIs change only at a particular point of time, and
– check if progress reporting is done continuously or periodically.

The change in a KPI was defined as an event when a stakeholder actively updated the status of the KPI, which means that there is an update event in the database where the KPI status was updated. Process-wise, this corresponds to the situation when a stakeholder needs to make an active assessment of the value of the KPI. He/she sees a notification on his dashboard and should make an active choice (even if it is only to confirm that the status is the same (e.g. still "green"). This means that the project manager at a higher level has confidence that KPIs status is up-to-date.

A co-change was defined as the update of two KPIs in the same period of time – in this case during one day.

These patterns are used for further analyses of co-changes and quantifying these changes as percentage. The dependencies are identified using the method developed in the previous research of the authors to monitor co-dependencies in software modules [21], using the co-change model presented in Figure 4.

The figure presents the lifeline of one assignment (e.g. a project) with the changes in KPIs. Two KPIs are considered as potentially dependent on one another if they change within the same day in the majority of days. For example, if KPI-A changes 10 times during the period

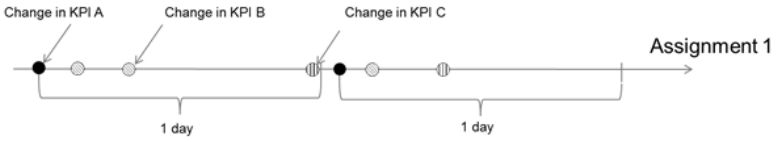


Figure 4. KPI dependency analysis model

of 10 days (once a day), and KPI-B changes 8 times during the period of the same 10 days, then these two KPIs can be considered as 80% dependent. This kind of analysis allowed to identify superfluous KPIs and to understand potential dependencies between them.

After calculating these dependencies in a single project, the dependencies were summarized for all studied projects. In this way, dependency pairs were obtained for all projects. In order to sort the changing KPIs, the PRE(x) function, which is defined as a percentage of the projects which have a co-dependency of strength x or more, was used. However, there could be cases when a KPI is available only in one or two projects. In such a case the AV(x) function was used as another criterion. The AV function is defined as the percentage of the projects where a KPI is used. For the analysis purposes PRE(75) and AV with the cut-off point of 33% (dvs. KPIs present in at least 33% of the projects) were used.

The data from the interviews was analysed by the main author of this study using coding. The results were discussed with the reference group for the project consisting of the other co-authors, two line managers and two experts at the company.

4. Results

This section presents study findings structured by research question.

4.1. How are KPIs defined?

The definition of KPIs consists of two parts – the measurement method describing how to collect the data for a given KPI and the decision criteria describing how to set the colour of the KPI (red, yellow or green). This means that the definition corresponds to the groups of activities prescribed by the theoretical framework adopted in this study (Section 2 – extraction and analysis). The KPIs are visualized using a web portal, as presented in Figure 5.

It was found that the measurement method for the KPIs could be defined in two ways:

- measuring that an activity has been performed (digital answer yes or no) – for example that a review of requirements has been performed, or
- counting the number of elements of a given type – for example how many defects of a specific type were discovered

These two measurement methods correspond to two models M-A and A-M. Both include the evaluation source systems before reporting a KPI – one is done automatically by extracting information (counting the number of elements) and the other one is the measurement that an activity has been performed (digital answer).

It was have found that these two measurement methods (and thus the reporting models) are used interchangeably, but their frequency changes over time. In the early stages of the project, it is common to use KPIs measuring the performance of an activity. In the late stages of the project it is more common to use KPIs representing the count of product elements – how ready the project is for release.

During the interviews, it was found that this transition from the reporting of activity progress to the reporting of product readiness is common throughout the project. Counting a number of elements is used for product-related KPIs, as it is easier to count elements that are "ready" or "tested" towards the end of the project (where the product becomes more tangible for the project). The process related KPIs showed that an activity was performed and therefore they were more common towards the beginning of the project.

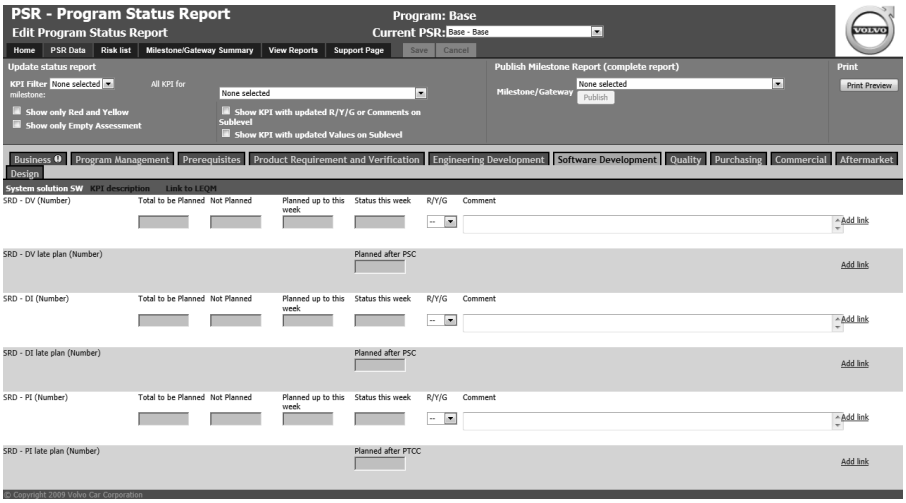


Figure 5. Web interface for status reporting

An example of the process-related KPI, which is calculated by measuring whether an activity has been performed, is the Requirement reviewed KPI for one of the early milestones in the project. The KPI has a colour set to green when all requirements are reviewed, to yellow when not all requirements are reviewed but there is a plan how to review all requirements and red when not all requirements are reviewed but there is no plan how to achieve them.

An example of the product-related KPI is the software product quality KPI. The KPI is calculated by counting defects which have a certain severity. The rules for setting the colours of the indicator depend on the phase of the project. Although it is calculated only in the last milestones of the project, the criteria for setting the values include the severity of defects and the number of defects. A criterion for setting red at one of the milestones for this indicator is:

– Green: number of defects with Severity 1, 2, 3, and 4 is 0 in status New or Open.
– Yellow: not meeting the target but with agreed plan in place to achieve target.
– Red: number of defects with Severity 1, 2, 3, and 4 is more than 0 in status New or Open or any (Severity 1, 2, 3, and 4) Passed Requested Target series.

The second type of KPIs – based on counting the number of the elements of a specific kind – are more quantitative in nature and that criteria for setting the colours (levels) of the KPI are clearer than for the first type of KPIs – based on measuring that an activity has been performed.

The interviews allowed to establish that the KPIs of these two kinds are mixed and that there is a need for more alignment. The interviewees also mentioned that having both types of the KPIs makes it difficult to visualize the status of the project at a specific moment – as some of the "greens" never change (performance of an activity) and some of the "greens" might change over time (number of defects).

4.1.1. An example of a product-related KPI – Software Product Quality

This KPI uses a defect tracking database as the source system. The summary of both the number of KPI updates per week (the top chart in the figure) and the data in the source system (the bottom chart in the figure), is presented in one diagram in Figure 6.

The colours of the bars in the top chart indicate the colour of the KPI reported. The colours of the bars in the bottom chart in the figure indicate the different status of the open defects in the database. The lines in the bottom chart show the cumulative number of defects reported in the entire project.

The criteria for setting KPI colours are related to the timing and the number of open

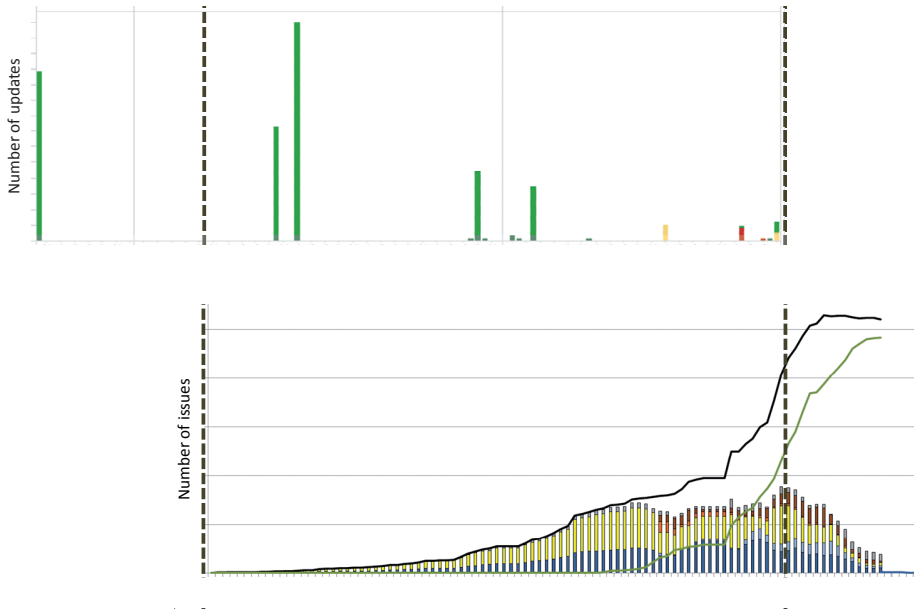


Figure 6. Comparison of updates of KPI and the underlying data from source system (defect reporting database). The vertical lines show alignment in time. Due to the sensitivity of the data the scales have been removed

(non-resolved) defects. For example, in order to set the status to yellow, there should be at least one newly reported defect after a specific milestone in the project. In order to set the status to red, the newly reported defects have to come after yet another (later) milestone. This is the case in this pilot project and our interviews have confirmed that the colours of the KPIs for this indicator are indeed set based on these criteria.

Since these criteria are so well defined, in terms of measurable quantities (number of defects in a specific status and milestone), KPI reporting could be supported by the pre-setting of the status of the KPI; thus, reducing the burden of searching for data for the sub-project managers.

4.2. How often are the KPIs reported in practice?

In order to study the patterns of KPI changes per week, a contingency table which summarized the number of KPI changes per week was calculated. Their visualisation was made using heatmaps, as it is shown in Figure 7. As the figure shows there are visible "vertical" lines where a set of KPIs changes the status. This indicates that the project management focuses on "milestones" when reporting KPIs to the database. This finding was also confirmed in the interviews. The reasons for this can be the lack of use of the KPIs between the milestones, and the need to prioritize other assignments.

During the interviews, it was also found that, given the non-continuous update of KPIs, it is difficult to obtain the overview of the current status. If a vertical line is drawn in the figure – a snapshot, it is not clear how "old" the status of each of the KPIs is.

To summarize the data, a histogram of the percentage of KPIs that change over one week was used, as presented in Figure 8.

Figure 9 shows the frequencies of KPI changes per week. Each row represents a project and each column represents the percentage of KPIs that changed per week. Each bar represents the number of weeks when a given percentage of KPIs changed.

As shown in the figure, there is no project where over 60% of KPIs changed and in the majority of weeks less than 10% of KPIs were changed. This shows that model A is not applied at the company as it is characterized by the continuous update of KPIs. During the in-

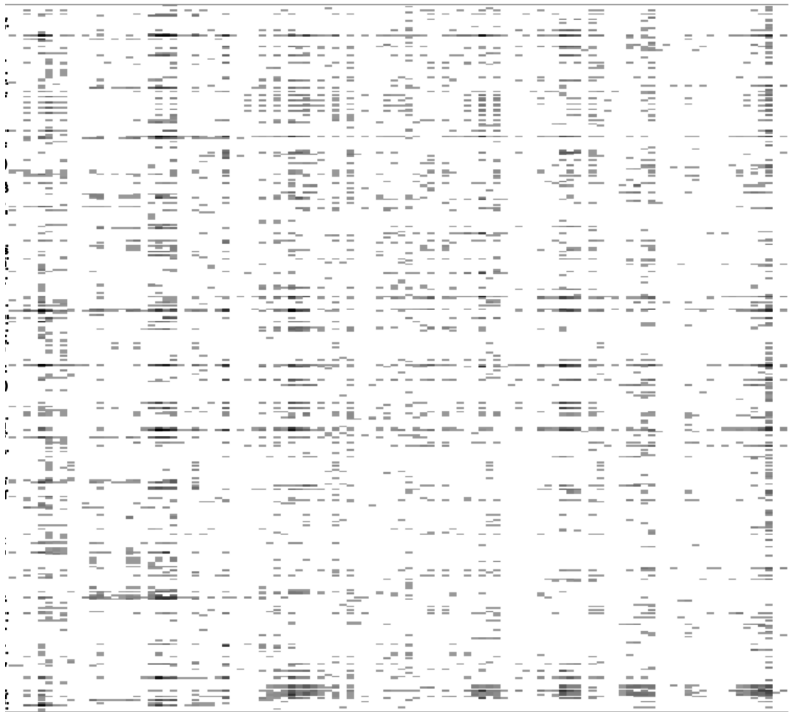


Figure 7. Frequency of reporting KPIs per week. Each row is one KPI and each column is one week; the intensity of the color indicates how often the KPI was updated during given week

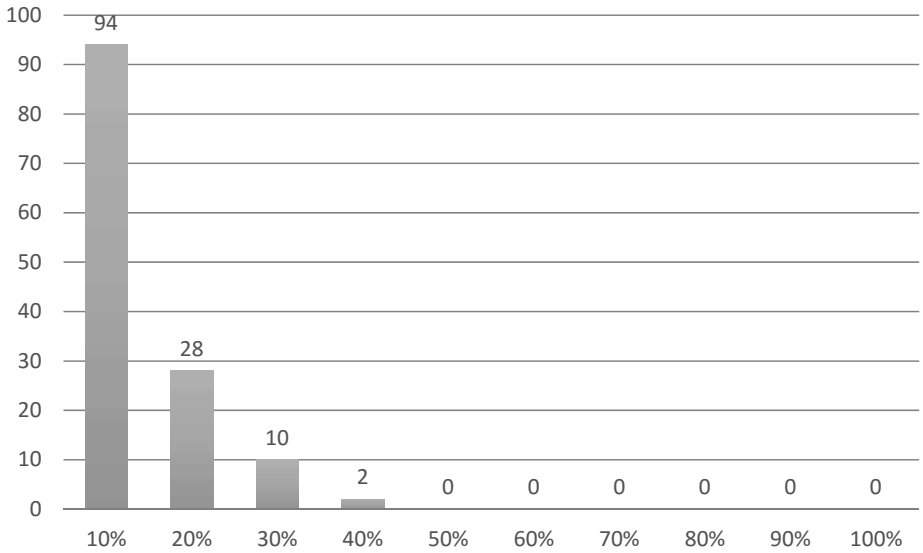


Figure 8. Histogram over the frequency of changes per week – each bar represents the number of weeks when the given percentage of KPIs changed

terviews it was found that model M is not applied either, as KPIs are calculated based on the data from source systems (e.g. requirements database, project planning tools). The interviewees explained that, depending on the indicator, they apply either model M-A or A-M.

At the beginning of the project, it often occurs that model M-A dominates, as the KPIs used at the beginning focus on tracking activities; whereas towards the end of the project it is model A-M which dominates, as the product data is often used for KPI calculation.

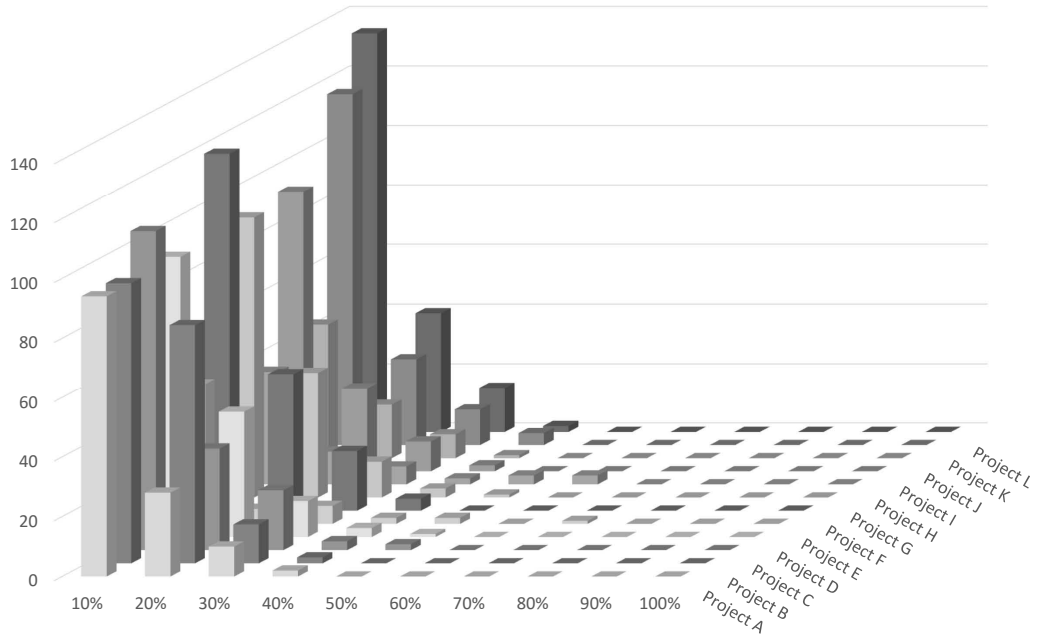


Figure 9. Frequencies of KPI changes in all studied projects

The interviews showed that this milestone-orientation was common – all interviewees supported that. They also indicated that one always needs to comment on the "red" status and therefore the list of risks is an important complementary tool. Each risk can be linked to a KPI and the risks are discussed during project status meetings.

The interviewees showed that, in practice, model A-M was challenging as it was difficult to assess the status of continuously changing data. In the cases of continuously changing data, one needs a fully automated model A to keep the pace of status assessment.

Another challenge identified, which requires the automated model A, is the fact that data import from source systems can be out-of-sync due to the large complexity of the source systems. Interdependencies between systems make it difficult to import all the data at once, and therefore the imports have varying frequency – which makes it difficult to make manual assessments (one does not know exactly how "fresh" the numbers are, i.e. has the data of low timeliness).

4.3. Who is responsible for reporting and acting upon KPI definitions?

The interviews showed that KPIs are reported by project managers and their sub-managers. However, they are always approved by the line managers or the project management team.

The sub-project and unit project managers are responsible for the assessment of KPIs (setting the colour based on the criteria) and for presenting them for the main project manager. When they make the assessment of a KPI, they present their assessment to the line managers of their respective unit or group. The manager approves or adjusts KPI assessment. It was found that it was formally the manager (in all cases), who was responsible for a given KPI.

After the approval of a KPI by the manager, it is the sub-project or unit project manager (depending on the level), who presents the KPI

to the subsequent higher level. Sub-project managers present KPIs to unit project managers and the unit project managers present KPIs to the main project manager.

The total project status is presented by the main project manager to a project steering group. The project steering group is the outmost responsible body for the project and can make such decisions as increasing the funding of the project, if needed.

4.4. Which KPIs are co-dependent on one another?

In this analysis the co-change analysis was used. When presenting the results, however, KPIs are listed based on two parameters according to the research methodology described in Section 3:

- PRE(x) function which is defined as a percentage of projects in which the co-dependency of strength x or more occurs, and
- AV function which is defined as a percentage of projects where a KPI is used.

Together with the company stakeholders, the authors identified the thresholds for these two functions to be: PRE(75) = 100% (meaning that in all projects where the KPI pair is present the strength of dependency is 75%) and AV = 33% (meaning that a KPI pair is present in at least 33% of the studied projects, i.e. 4 projects). The application of these functions to the data set (over 244 000 pairs of KPIs) showed that there were 75 pairs that were manually reviewed by the stakeholders to establish if they overlapped. Eight (8) pairs were found to be dependent on each other, 27 pairs were assigned for further investigation to test if the statistical dependency can be confirmed by experts, 3 pairs were already removed (after the end of the projects and before the end of the study) and 37 pairs were found to be false positives. The 37 KPIs which were found to be false positives were present in only 4 projects, whereas the eight pairs found to be co-dependent were found in all 12 studied projects. The 27 pairs assigned for further investigation were found in between 5 and 9 projects from the 12 projects sample.

The interviews revealed that the majority of the KPIs in the list of 75 KPIs were progress indicators. These progress indicators were used to indicate that sets of activities were successfully conducted based on the set of pre-defined quality criteria.

During the interviews, it was found that the KPIs which are related to a specific deadline were usually reported in the period of four weeks before the milestone until one week after the milestone. The four weeks period allows the project (and subproject/unit project managers) to focus on the goal and report the KPIs which are important for assessing the completion of the milestone.

4.5. Is there a difference between software-related KPIs and non-software-related KPIs?

This analysis, allowed to identify which KPIs were software-related, apart from this a correlation analysis between the software-related and non-software-related KPIs was conducted. The correlated elements were the trends in the reporting of these KPIs. As a result, 20 KPIs to be related to software development activities were identified. Once these software-related KPIs were identified, they were added to the changes so as to create a time series for these changes. Then the same analysis was performed for the other, non-software-related KPIs. An example of this chart is presented in Figure 10.

The diagram indicates that the changes in the KPIs per week follow the same trend. In particular, the visual analysis shows that the peaks in the number of KPIs reported happen at the same time. The Pearson correlation coefficient for these two series is 0.69, which is a strong correlation.

It can be observed that the peaks in the number of KPI changes is similar. However, it can also be observed that the software-related KPIs have more peaks. This can be explained by the fact that the software-related KPIs have higher similarity to each other than non-software-related KPIs. The non-software-related KPIs contain

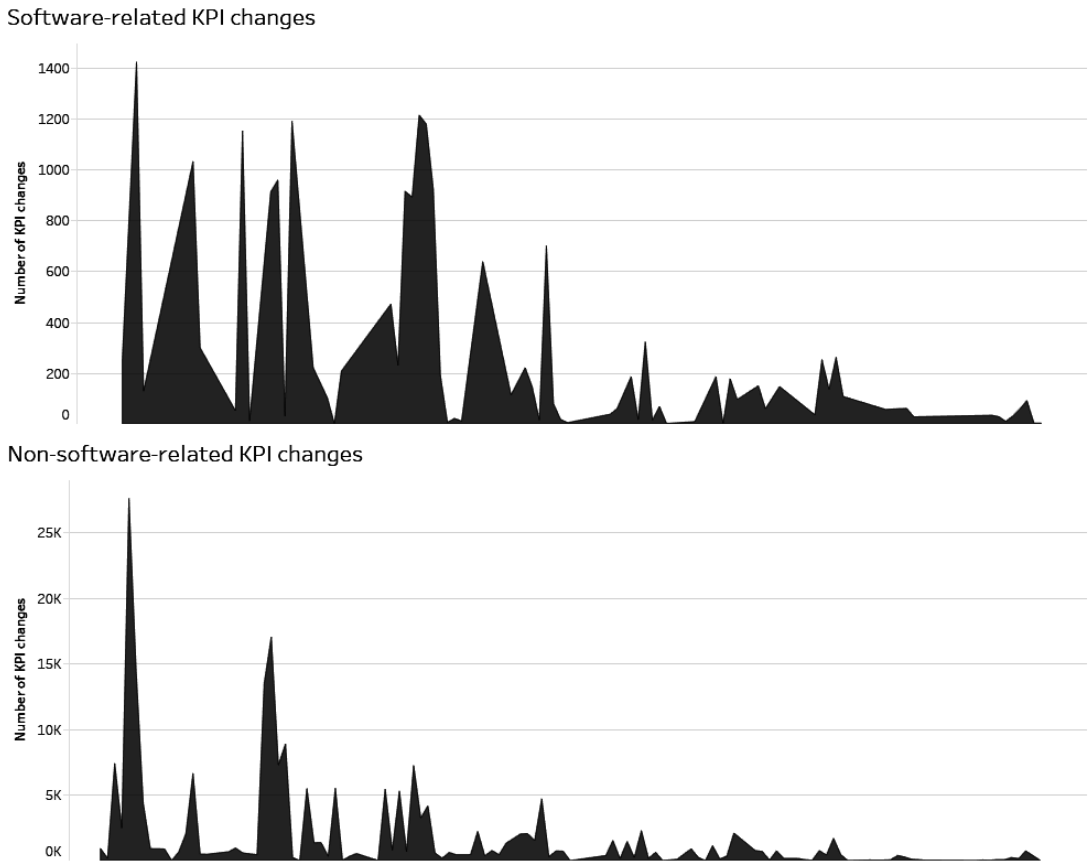


Figure 10. Changes in software-related and non-software-related KPIs per week

all other disciplines (except for software), which explains the lower similarity in that group.

Table 2 presents the results of the correlation analysis for all studied projects.

Table 2. Pearson correlation coefficients for time series of number of changes in KPI values for software-related vs. non-software-related KPIs

Project	KPIs	Correlation
Project A (pilot)	386	0.69
Project B	552	0.69
Project C	396	0.65
Project D	382	0.73
Project E	257	0.86
Project F	442	0.83
Project G	305	0.73
Project H	252	0.72
Project I	421	0.61
Project J	342	0.63
Project K	494	0.84
Project L	431	0.86

The table shows high correlations, which means that the trends are similar. This in turn means that the reporting of software-related KPIs is similar to the non-software-related KPIs.

4.6. Summary – How are KPIs for monitoring project progress used in practice in a large product development organization?

Based on the results, the following trends in KPI use were observed:

1. The definition is standardized using three levels – red, yellow, green.
2. The trend in reporting software-related KPIs is similar to the trend in reporting non-software-related KPIs.
3. The definition of each of the levels is based on the ability to act upon the problems indicated by the KPI – if the status is yellow, then the

responsible project manager has a plan how to get back on track before the next milestone; if the level is red, then the project manager registers the risks in the qualitative description of risks.

4. There is some progress from process-oriented to product-oriented KPIs over the project time – at the beginning of the project, when no product is ready yet, the control of KPIs whose activities have been performed (e.g. requirement reviews are completed). At the end of the project, when there is a product, KPIs monitor if the product is ready for release (e.g. that all ECUs are tested to 100%).
5. The reporting of KPIs is done based on milestones – unlike the standard software development projects, KPIs are used to check whether a project has achieved the degree of readiness required to move to another phase (e.g. whether all software components for an ECU are ready to be tested in a simulated environment).
6. The responsibility for the development of a KPI is given to measurement champions who have the necessary knowledge about the domain and the specifics of the product.
7. The responsibility for the reporting of KPI value is given to the project managers at different levels as they are responsible for ensuring that the product and project follow the set resources.
8. The responsibility for the formal approval of the KPI status is given to the line managers of relevant organizations, as they have the responsibility for resources in a company.

The above items show that project management in the automotive sector, including embedded software projects, follows the classical, gate- and milestone-oriented principles. According to these principles, planning and following the plan are central.

The fact that the number of KPIs is between 252 and 552 shows that the complexity of the development is too high to change to the continuous reporting of KPIs. Therefore, there is a need to combine manual assessment with automated data provisioning from source systems.

4.7. Validity analysis

In this paper the framework of Wohlin et al. [22] and a more recent framework by Runeson et al. [17] were followed in evaluating the validity of the research.

The main threat to the external validity is the fact that only one company is studied – Volvo Car Group. Single-case studies can risk bias towards informants or the context. However, the authors believe that a broad sample of projects helps to increase the external validity at least beyond one project. This study is also considered an important contribution to the studies of software development as part of a larger context – car development.

The main threat to the construct validity is the potential mono-operation bias as the study was conducted at a single company only. In order to minimize this threat by diversifying the project sample – the authors chose a sample of 12 projects with different characteristics, scopes and project teams. Another threat which was minimized is the mono-method bias – using a single measure of co-dependency. It was minimized by using the same measure as in the previous studies where the validity of finding co-dependencies was established (although for a different type of a measured entity). Finally, another identified threat related to the measurement of co-change was the fact that some co-changes are purely random. To minimize the threat, the PRE(x) and AV(x) functions were used to set thresholds to reduce the probability of capturing a random change as a valid co-change. It was also taken into account that the stakeholders do not update the status of a KPI when the value changes and the status remains the same (e.g. "red"). This is a threat that could not be minimized as there was no measurement method to capture this situation, however, it was established that this situation seldom occurred in the studied company.

Using the theoretical model, shown in Figure 1, the study was constructed in such a way that each answer was coded based on the elements of the model. This introduced the risk of missing important information, which was limited when using alternative methods (e.g. grounded theory). However, the model was cho-

sen as interviews were to be used as a triangulation method to the quantitative data analysis.

In this study the risks related to the internal validity and reliability were to be minimised. Therefore, the triangulation of data collection methods, i.e. statistical and archival analyses (KPI data from the database) and interviews, was used. The triangulation allowed to check the root causes of the patterns visible in the statistical analyses – e.g. identifying potential causes of KPIs being co-dependent.

In general the main threat to the conclusion validity of such studies as the one presented in this paper – identifying co-dependencies in numerical data sets – is drawing conclusions based only on statistics. Thus, it was decided to triangulate the data sources (document analysis and interviews) and these triangulated interviewees and their roles (multiple organizations within Volvo Car Group and multiple roles) were used. The triangulation also minimized the risk that the results would be biased or would represent a specific part of the organization.

Since interviews were used in the study, the authors are aware of the conclusion validity threats related to overinterpretation. In order to minimize these threats, the findings were confirmed during a workshop with all the interviewees and in another workshop with a reference group for the project (consisting of technical experts and managers).

5. Related work

In a recent study Todorovic et al. [23] explored the types of KPIs which can be found in organizations focused on projects. They found, which was confirmed in this study, that there are such types of KPIs which are related to progress and such that are related to performance and that the latter are more challenging. It can be thus concluded that the theoretical models, published in software engineering literature about KPIs, need extensions to improve the formulas by adding uncertainty components or temporal aspects, or both. In a recent study, Todorovic et al. [23] identified the properties of KPIs which are important for this study, too – a KPI being actionable and measureable. In their study Todorovic also postulated the fact that there is a difference between measures and KPIs which corresponds to a similar difference between progress KPIs and performance KPIs. The latter are naturally important, but as Todorovic et al. state, also very challenging because in the project oriented organizations the progress KPIs tend to be the most common ones. However, even though it could be observed that in practice the theoretical framework of Todorovic et al. can be applied, their models, however, cannot because the models are based on the assumptions that there is a known and updated status at the moment when the model is applied.

Pilgoret [24] lists a number of KPIs which are important for modern project managers. The list includes both process and performance KPIs, but does not show when and how the KPIs should be reported. In this study new evidence showing that not all KPIs are equal and that they are reported only when necessary is provided. The evidence that KPIs are not reported continuously and at the same time they are reported more than once per project means that there is a change in the way in which KPI are traditionally used. An example of such a study is the study of Pilorget [24] who shows the estimations of a number of KPIs (although much smaller than the number found in our study) used only once during the project. Since the KPIs change and get updated, methodologies like this need to be adjusted to support more continuous re-evaluation of the project status value and should include the temporal component – how "old" or "stable" the data in the KPIs is.

Colin and Vanhoucke [25] presented a recent study on statistical performance control approaches for project monitoring. The results recognized the challenges with continuous monitoring of project status with KPIs, which is addressed in this work.

The use of KPIs is very common in the field of corporate performance measurement and the predictions are that it will become even more important[3]. In the interview with the main experts behind the Balanced Scorecard approach, this trend becomes even more evident. Therefore,

in this study we intended to explore the use of KPIs in practice, and explore what the industrial trends in this area are.

In a study by Jaafari [26], the challenges with milestone-based project reporting were recognized. The evaluation of the presented PH-Check approach in the industrial context showed that the milestone-based assessment leads to efficient identification of shortcomings, and provide the ability for stakeholders to react. This positive view of the milestone-based reporting is shared by some of the interviewees in the presented study.

As the modern product development evolves, so do the practices related to it. In a recent study of how the golden triangle in project management is perceived on the lowest levels of the organization, Drury-Grogan [27] found that quantitative information is of less value for lower levels. Instead, the consensus and understanding of project goals become more important as the responsibilities of development teams increase. For cross-functional software development teams, as Drury-Grogan found, the product is of more focus than the project. This study also aimed at understandnding whether these findings were valid for car development projects, too.

The rationale behind this study was to follow up on the study of Steyn and Stoker [4], who found that there was a significant difference between the performance of projects, depending on which project measurement methodology was chosen. They found that measuring such parameters as contingency task allowance can increase project performance. Although limited to a small number of measures, the study shed light on the use of measurement methodology which can make a difference in projects. Therefore, it was decided to study the practice of how the measures (in this case KPIs) are used and when they are reported. In the light of the research of Steyn and Stoker [4], it could be concluded that it is perhaps not the use of measures as such but the use of measures related to organizational goals (in the same sense as Todorovic et al.'s study) can make a difference. It was found that performance oriented KPIs stimulate goal orientation and can be (in principle) updated continuously which increases the probability of project success. This means that in project management measurement the separation of these two concerns – progress and performance oriented KPIs reporting should be postulated.

Raymond and Bergeron [28] surveyed 39 project managers with regards to the effects of the use of reporting systems in project management. The results showed that the use of project management information systems improved the efficiency and effectiveness of managerial tasks in projects. These results were important as an input to discussions with the industrial partners in this study, and were confirmed in this research – these kinds of information systems provide value. However, it was found that it is important to find the right balance between the cost of reporting and the value obtained.

Marques et al. [29] presented a method for aggregating the status of KPIs (and metrics) in complex products. Their case study illustrated how feasible aggregation is in practice. Marques et al.'s approach is an alternative to the use of multiple KPIs at the total project level, which is the case in the studied company.

In our previous work, the use of KPIs was studied in software development organizations, exemplified by Ericsson [30], [31]. The findings showed that the number of KPIs can be small and that the automation is the key aspect. The results from the study, presented in this paper, contradict the results from the previous studies – lower degree of automation and larger number of indicators. This can be explained by the fact that the previous studies investigated software development projects, whereas this study investigated both software and non-software development projects, as car development projects combine significantly more disciplines than software development projects.

Sanchez and Robert [32] provided a framework for defining KPIs, where they combine standard KPIs, such as the earned-value indicator with indicators related to risk management. As a result of the analysis of the pattern of reporting risks in the studied projects, this case study provides evidence that this kind of combination is very much needed, as the risk-view of the KPI provides a more complete view on project performance.

Choosing a set of metrics and KPIs for monitoring f projects was also studied in the Netherlands [33], with results showing that the most important determinant of the success of KPIs adoption is the behaviour of the organization. The studied organizations identified a gap between the research and development function and the R&D department's responsibility, which can be observed in the studied projects – the discrepancy between the process and project.

The quality of KPIs was also studied in a number of contexts, and, as a result, an interesting study was by Spangenberg and Göhlich [34], who studied how to construct the roadmaps of mechatronic software systems using KPIs. In their context of transportation systems, technology readiness levels were combined with goal-oriented KPIs, such as carbon dioxide emissions, in order to allow the simulation of project outcomes. The presented study provides evidence that this kind of goal needs a change in the mindset of project management, as KPIs are progress oriented at the beginning of the project. However, this kind of simulations are theoretically possible towards the end of the project, where the focus shifts towards product-oriented KPIs.

Lainhart et al. described the methodology for software project governance – COBIT, [35] – which provides a holistic approach to managing software projects. Our work provides evidence that milestone-oriented project monitoring (as described in the COBIT processes part) is an important industrial practice.

Finally, this work can be considered an input to planning measurement programs using frameworks, such as GQM+Strategies, [36], [37]. They help to establish and evolve measurement programs by defining KPIs and relating them to company strategies. Our findings, that KPIs are used when assessing milestones, can be an input to defining KPIs which are to be used in the way familiar to the automotive software engineering industry.

6. Conclusions and further work

This study explored the question of how KPIs for monitoring project progress are used in practice in a large product development organization. The study encompassed the analysis of how KPIs are used in practice by investigating 12 car development projects at the Volvo Car Group. Interviews and statistical co-dependency analyses were used to explore how many KPIs exist and how often they are reported. The reference for the analysis were four theoretical models.

The results show that the number of KPIs used in projects oscillates between 252 and 552 and captures the complexity of the product as well as the complexity of the project to develop it. Over time the projects change the focus from activities (i.e. what has been done in the project) to product readiness (i.e. what needs to be done in order for the product to be ready for launch). The number, diversity, frequency of updates and the change of focus over time show that the studied company is very mature in the use of KPIs. The results from these investigations supported the company in identifying KPIs which can be dependent on one another (a situation quite possible in such a large data set).

The KPIs for embedded software development are correlated with other KPIs. This implies that, regardless of the applied software development methodology, progress reporting can follow the standard, milestone-oriented progress reporting. It means that companies have some flexibility in changing their ways-of-working within the given frames of strict non-software development projects.

Our further work is an in-depth study to optimize the set of reusable KPIs for an embedded software development project, and to evaluate its feasibility on more cases. This optimal set would help managers to benchmark the projects against each other, quantifying the performance of the project portfolio over time.

Acknowledgements

We would like to thank the Volvo Car Group for letting us conduct the study. We are also grateful to the anonymous reviewers for advice and help in improving the paper.

The research has been partially sponsored by the Swedish Strategic Research Foundation, under grant number SM-13-007.

References

[1] W.S. Humphrey, *Managing technical people: innovation, teamwork, and the software process*. Addison-Wesley Longman Publishing Co., Inc., 1996.

[2] M. Staron, W. Meding, J. Hansson, C. Höglund, K. Niesel, and V. Bergmann, "Dashboards for continuous monitoring of quality for software product under development," *System Qualities and Software Architecture (SQSA)*, 2013.

[3] A.A. De Waal, "The future of the balanced scorecard: an interview with Professor. Dr Robert S. Kaplan," *Measuring Business Excellence*, Vol. 7, No. 1, 2003, pp. 30–35.

[4] "Does measurement theory impact project performance?" *Procedia - Social and Behavioral Sciences*, Vol. 119, 2014, pp. 635 – 644.

[5] A. Neely, B. Marr, G. Roos, S. Pike, and O. Gupta, "Towards the third generation of performance measurement," *Controlling*, Vol. 15, No. 3/4, 2003, pp. 129–135.

[6] International Standard Organization and International Electrotechnical Commission, "ISO/IEC 15939 – software and systems engineering, software measurement process," ISO/IEC, Tech. Rep., 2007.

[7] R.S. Kaplan and D.P. Norton, "Putting the balanced scorecard to work," *Performance measurement, management, and appraisal sourcebook*, Vol. 66, 1995, p. 17511.

[8] M. Staron, W. Meding, J. Hansson, C. Höglund, K. Niesel, and V. Bergmann, "Dashboards for continuous monitoring of quality for software product under development," *System Qualities and Software Architecture (SQSA)*, 2014.

[9] R.S. Kaplan and D.P. Norton, *The balanced scorecard: Translating strategy into action*. Harvard Business Press, 1996.

[10] C. Bentley, *Practical Prince2*. The Stationery Office, 2005.

[11] C. Fornell, "A national customer satisfaction barometer: The Swedish experience," *Journal of Marketing*, 1992, pp. 6–21.

[12] A. Assila, K. Marçal de Oliveira, and H. Ezzedine, "Integration of subjective and objective usability evaluation based on ISO/IEC 15939: A case study for traffic supervision systems," *International Journal of Human–Computer Interaction*, Vol. 32, No. 12, 2016, pp. 931–955.

[13] M. Staron, K. Niesel, and W. Meding, "Selecting the right visualization of indicators and measures–dashboard selection model," in *Software Measurement*. Springer, 2015, pp. 130–143.

[14] M. Staron, W. Meding, and C. Nilsson, "A framework for developing measurement systems and its industrial evaluation," *Information and Software Technology*, Vol. 51, No. 4, 2008, pp. 721–737.

[15] A.A. Mughal, *A Framework for Implementing Software Measurement Programs in Small and Medium Enterprises*, Ph.D. dissertation, University of Otago, 2017.

[16] M. Staron, "Critical role of measures in decision processes: Managerial and technical measures in the context of large software development organizations," *Information and Software Technology*, Vol. 54, No. 8, 2012, pp. 887–899.

[17] P. Runeson, M. Host, A. Rainer, and B. Regnell, *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons, 2012.

[18] B. Azvine, Z. Cui, and D. Nauck, "Towards real-time business intelligence," *BT Technology Journal*, Vol. 23, No. 3, 2005, pp. 214–225.

[19] M. Staron, J. Hansson, R. Feldt, A. Henriksson, W. Meding, S. Nilsson, and C. Hoglund, "Measuring and visualizing code stability–a case study at three companies," in *Software Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2013 Joint Conference of the 23rd International Workshop on*. IEEE, 2013, pp. 191–200.

[20] R. Feldt, M. Staron, E. Hult, and T. Liljegren, "Supporting software decision meetings: Heatmaps for visualising test and code measurements," in *Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on*. IEEE, 2013, pp. 62–69.

[21] M. Staron, W. Meding, C. Hoglund, and J. Hansson, "Identifying implicit architectural dependencies using measures of source code change waves," in *Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on*. IEEE, 2013, pp. 325–332.

[22] C. Wohlin, P. Runeson, M. Host, M.C. Ohlsson, B. Regnell, and A. Wesslèn, *Experimentation in Software Engineering: An Introduction*. Boston MA: Kluwer Academic Publisher, 2000.

[23] M. Todorović, Z. Mitrović, and D. Bjelica, "Measuring project success in project-oriented organizations," *Management*, Vol. 68, 2013, pp. 41–48.

[24] L. Pilorget, "Process performance indicators and reporting," in *Implementing IT Processes*. Springer, 2015, pp. 177–197.

[25] J. Colin and M. Vanhoucke, "Developing a framework for statistical process control approaches

in project management," *International Journal of Project Management*, 2015.

[26] A. Jaafari, "Project and program diagnostics: A systemic approach," *International Journal of Project Management*, Vol. 25, No. 8, 2007, pp. 781–790.

[27] M.L. Drury-Grogan, "Performance on agile teams: Relating iteration objectives and critical decisions to project management success factors," *Information and Software Technology*, Vol. 56, No. 5, 2014, pp. 506–515.

[28] L. Raymond and F. Bergeron, "Project management information systems: An empirical study of their impact on project managers and project success," *International Journal of Project Management*, Vol. 26, No. 2, 2008, pp. 213–220.

[29] G. Marques, D. Gourc, and M. Lauras, "Multi-criteria performance analysis for decision making in project management," *International Journal of Project Management*, Vol. 29, No. 8, 2011, pp. 1057–1069.

[30] M. Staron and W. Meding, "Transparent measures: cost-efficient measurement processes in SE," in *Software Technology Transfer Workshop, Kista, Sweden*, 2011.

[31] M. Staron, W. Meding, and K. Palm, "Release readiness indicator for mature agile and lean software development projects," in *Agile Processes in Software Engineering and Extreme Programming*. Springer, 2012, pp. 93–107.

[32] H. Sanchez and B. Robert, "Measuring portfolio strategic performance using key performance indicators," *Project Management Journal*, Vol. 41, No. 5, 2010, pp. 64–73.

[33] J. Bilderbeek *et al.*, "R&d performance measurement: more than choosing a set of metrics," *R&D Management*, Vol. 29, No. 1, 1999, pp. 35–46.

[34] F. Spangenberg and D. Göhlich, "Technology roadmapping based on key performance indicators," in *Smart Product Engineering*. Springer, 2013, pp. 377–386.

[35] J.W. Lainhart IV, "COBIT^{TM}: A methodology for managing and controlling information and information technology risks and vulnerabilities," *Journal of Information Systems*, Vol. 14, No. s-1, 2000, pp. 21–25.

[36] V. Basili, J. Heidrich, M. Lindvall, J. Munch, M. Regardie, and A. Trendowicz, "GQM+ Strategies–Aligning Business Strategies with Software Measurement," in *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*. IEEE, 2007, pp. 488–490.

[37] J. Münch, F. Fagerholm, P. Kettunen, M. Pagels, and J. Partanen, "Experiences and insights from applying GQM+Strategies in a systems product development organisation," in *Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on*. IEEE, 2013, pp. 70–77.

e-Informatica Software Engineering Journal, Volume 12, Issue 1, 2018, pages: 237–263, DOI 10.5277/e-Inf180110

Semantic Knowledge Management System to Support Software Engineers: Implementation and Static Evaluation through Interviews at Ericsson

Ali Demirsoy*, Kai Petersen**

*Borsa Istanbul

**Fachbereich Wirtschaft, Department of Software Engineering, University of Applied Sciences Flensburg, Blekinge Institute of Technology

alidemirsoy@gmail.com, kai.petersen@bth.se

Abstract

Background: In large-scale corporations in the software engineering context information overload problems occur as stakeholders continuously produce useful information on process life-cycle issues, matters related to specific products under development, etc. Information overload makes finding relevant information (e.g., how did the company apply the requirements process for product X?) challenging, which is in the primary focus of this paper.

Contribution: In this study the authors aimed at evaluating the ease of implementing a semantic knowledge management system at Ericsson, including the essential components of such systems (such as text processing, ontologies, semantic annotation and semantic search). Thereafter, feedback on the usefulness of the system was collected from practitioners.

Method: A single case study was conducted at a development site of Ericsson AB in Sweden.

Results: It was found that semantic knowledge management systems are challenging to implement, this refers in particular to the implementation and integration of ontologies. Specific ontologies for structuring and filtering are essential, such as domain ontologies and ontologies distinct to the organization.

Conclusion: To be readily adopted and transferable to practice, desired ontologies need to be implemented and integrated into semantic knowledge management frameworks with ease, given that the desired ontologies are dependent on organizations and domains.

Keywords: knowledge management, information overload, case study, semantic web

1. Introduction

One of the main challenges for large-scale organizations is the high number of stakeholders [1]. They all provide/produce information and knowledge and, as a result, increase the amount of information. Besides, many stakeholders are not known to others as organizations grow; thus, the holders of specific pieces of knowledge are not known. Therefore, a significant problem occurs related to the communication and coordination between these stakeholders [2]. A solution offering assistance in overcoming these problems is knowledge management, i.e. the process of acquiring or creating knowledge, transforming it into a reusable form, and maintaining, finding and reusing it [3, 4]. Most of the current knowledge management systems use keyword-based search models that rely on words' lexical forms, rather than the meanings of the words [5]. However, these search mechanisms do not always satisfy the needs of users in terms of the precision of obtained results [6, 7]. In consequence, people who exchange information with each other face the problem of information overload due to the high number of available documents and information

[8–11], i.e. more relevant information than one can assimilate is available [12].

"Semantic Information Retrieval", also referred to as "Semantic Search" [13], has been proposed to address the information overload issue. Semantic search refers to retrieving information based on the interpretations of the meanings of words [6]. Traditionally, there are classical information-retrieval models [14] that are aimed to find the most relevant document for a given query. The models estimate the relevance of documents and rank them via probabilistic methods, such as the Bayes classifier model [15] and the vector space model [16]. However, these models retrieve textual information based on the words' lexical forms, not their meanings. Hence, there is a problem of many irrelevant search outputs as a result of the ambiguity of words. A word can have more than one meaning, or many words can describe the same meaning. In these cases, the results might be either irrelevant or insufficient [5, 7, 17]. There are also statistical approaches such as classifying and clustering, which are aimed to overcome these problems by relying on the statistical occurrences of the words [18]. These methods have been successful in some cases in increasing the hit rate during searching [19]. However, the semantic search goes one step beyond these approaches by enabling complex queries and retrieving extracted knowledge from the processed information sources. This way, users can search for meaningful queries instead of textual strings and, in addition to this, automated tasks can process information with a certain level of understanding [17].

There have been several studies that apply semantic technologies to the software engineering domain to conceptualize and organize the knowledge (e.g., [20–22]). These studies focused on different artefacts of the software development life-cycle (e.g., requirements and architectural assets). However, there are only a few examples that aim at organizing the existing knowledge to enhance knowledge reuse within a knowledge management system, where users share documents for the use of others [23,24]. These systems (e.g., blogs, forums, document repositories) are crucial to software engineers for utilizing the existing information by finding a relevant shared document and overcoming problems related to information overload [8, 25].

There is a lack of information how to implement and adopt semantic knowledge management solutions in an organization with no previous experience. Semantic knowledge management systems integrate different aspects (such as semantic annotation, querying, entity ranking, etc.) into an overall system. However, having an integrated solution also makes one less flexible as it is not so easy to simply exchange/expand ontologies as experienced in our study. Current research focuses on presenting final solutions and the ideas behind them but not the ways to make these solutions work [17, 26, 27]. Hence, there is a need to study the process of adopting semantic systems as the experience gathered from here would be valuable for similar adopters to understand the advantages, costs, and limitations of these systems. Given the high amount of information in documents, a more precise search possibility as well as a more natural way of annotating information could be useful, which is provided by semantic knowledge management systems. Though, for this to work, it must be feasible to implement and also be perceived as useful, which falls within the scope of this work.

Why to investigate a semantic approach as the information retrieval approach? The semantic approach not only offers solutions for achieving precision (number of relevant results compared to all retrieved results) and recall (number of relevant results compared to the number of results that ideally should have been found), but also provides extracted knowledge from the analysis of the contents of documents [28, 29]. Hence, it differs from all other models where the only aim is to retrieve the most relevant document. Here the objective is to retrieve the necessary knowledge, not the document or documents that contain this knowledge [28]. However, it can also be used to retrieve documents based on the semantics of documents and can be integrated with ranking techniques [7]. For this reason, the semantic web approach seems to be one step ahead of the other models, and the semantic search can be used to solve

the current problems in information retrieval. However, for machines to read, interpret and process the information one needs a syntactical model. Requirements on machine readability causes a limitation of the type of information which can be modelled and extracted from documents. The most important factor here is the context and the content of the documents, and also the form of the desired information in the documents. Hence, to use semantic search for solving information overload, the needs of the users concerning their information usage and the content of the documents for their domain have to be investigated and analysed to see if it applies to semantic information retrieval. For instance, using semantic technologies has been observed to be very useful in such areas as biology, since the modelled information in biology is very suitable to represent ontologies [30].

The primary goal of this work is to understand and evaluate the feasibility of the implementation of semantic knowledge management systems. The study makes two contributions:

- **Contribution 1:** After the investigation of the context of the company, a semantic knowledge management system was implemented, which highlighted the limitations of such systems from a feasibility perspective. Understanding the limitations is important as these may hinder the adoption in industry [31]. There is a definite need for solutions whose practice can easily implement and integrate into existing environments for a successful transfer to industry [32]. In the context of search-based software testing, Arcuri et al. [33] highlighted that search-based software testing is not readily transferable if no engineering efforts are taken; hence, to make it easy to integrate it and use with the existing systems in practice, additional engineering efforts are required. The ease of integration into existing solutions was a key factor for the successful transfer of research results to industry. The ease also determines the degree of evaluation which in turn is dependent on the degree of the readiness of the solutions available.
- **Contribution 2:** After that practitioners assessed the system by using it in the context of an interview session. The evaluation conducted was a static evaluation [34]. The tatic evaluation allows to gather early feedback in an exploratory fashion and to capture essential issues and needed corrections before further spreading and developing a solution. The evaluation provided valuable qualitative feedback on the potential of semantic knowledge management systems and about their strengths and weaknesses. This research presents a single case study [35] at the development site of Ericsson.

The research comprises three phases. First, the authors focused on understanding the research context. Second, they implemented a solution for the semantic knowledge management system and reflected on their experiences. Third, they conducted a static evaluation [34] gathering qualitative feedback on the solution proposed to identify the most crucial improvement suggestions.

The remainder of the article is structured as follows. Section 2 presents related work. Section 3 describes the research method. The results are presented in Section 4. The conclusions in Section 5 provide the answers to our research questions.

2. Related work

First basic terms concerning data, knowledge, and information are presented. After that, integrated semantic knowledge management frameworks are shown. The subsequent sections explain essential components (e.g., for information retrieval).

2.1. Data, knowledge, and information

Different definitions exist for data, information, and knowledge. According to Thierauf [36], data constitutes raw facts and figures. Data becomes information through contextualization and categorization. Documented experience and know-how already represent knowledge. Hence, documents produced in companies, such as the case company, may contain knowledge if they provide an experience report or a process of how

to solve a problem, however, they may also carry only information (e.g., product requirements) or data (e.g., sales figures).

2.2. Semantic knowledge management frameworks

Knowledge management systems are composed of various steps and corresponding tools. It requires a systematic methodology and considerable amount of time and expertise to extract and formalize knowledge from unstructured data and to develop a platform that can find, share and manage information. Hence, the authors will examine the research on knowledge management platforms that provide all these functionalities together. Semantic knowledge management systems introduce structure through ontologies, e.g. enabling faceted search where there is a browsable classification, making the structure of information explicit to the end user.

OntoShare: OntoShare is an organizational knowledge management system that promotes sharing of information between people who have mutual concerns or interests [37]. It is an ontology-based tool that places the profiles of the users at the centre of attention. That is, the interests of each user are modelled by an ontology and this information is extracted from the activities of a user. Every time the user shares some information, the system first performs a text analysis in order to extract the theme of the document, which will constitute a brief summary of the content. Then the system scans all other users' profiles in order to look for a strong match between the content of the document and the users' interests. When there is a relation which is strong enough, then the system emails the corresponding user to inform about the new document shared. Moreover, the content of the document is also compared to the author's interests in order to add new interests if necessary. OntoShare provides many semantic search capabilities as well as a keyword-based search supported semantically by the concepts and user profiles. The user can search for documents that they might be interested in, modify annotations of existing documents and also search for people that are interested in a certain area.

Knowledge and information management framework (KIM): KIM [27] is a platform for semantic annotation and semantic search over several kinds of information sources. It is used for information extraction from data pools based on an ontology and a knowledge base [27].

KIM comes with an upper-level ontology called PROTON which has about 300 classes and 100 properties in OWL Lite[1]. This ontology covers most general concepts, such as names of people, locations and organizations along with numbers and dates. It also has the KIM World Knowledge Base (WKB) which has about 200,000 entity descriptions to provide background knowledge for commonly known entities. KIM keeps the ontologies and the knowledge bases in the SESAME based Owlim[2] RDF(S) repository.

Moreover, KIM uses the GATE framework for information extraction processes and Lucene from Apache as a retrieval engine [38]. Lucene has been adapted so that it allows indexing by entity types and measure the relevance with respect to entity types.

KIM not only provides full-automatic semantic annotation, but also allows retrieving information based on the metadata that has been created. This brings a new perspective to information retrieval, as the user is able to define a "pattern search". That is, a semantic query can contain entities that are known or extracted before, relations between the entities and attributes of these entities [27]. This means the user can, for example, find out the names of the organizations in a specific location that have more than 100 employees in one single query. In this case, an organization would be an entity, a location and an employee number would be a relation and that specific location and the number 100 would be the attributes.

Semantic Wikis: Wikis are also a way used by large organizations to share all kinds of information and can be used for knowledge man-

[1]OWL Lite: https://www.w3.org/TR/owl-features/
[2]Owlim: http://graphdb.ontotext.com/documentation/7.0/enterprise/using-graphdb-with-the-sesame-api.html

agement. A Wiki is a hypertext environment that provides the collaborative editing possibilities of Web pages. Wikis emphasize openness, ease-of-use and modification [39].There are some limitations of Wikis that prevent them from being used as a knowledge management tool. Wikis do not provide structured access to data and do not support knowledge reuse [40]. A semantic Wiki provides annotation capabilities to create formal descriptions, retrieval mechanisms for semantic search, and semi-automatic meta-data extraction system to simplify the annotation process.

Active: The project Active [41] aims to increase the productivity of knowledge sharing via prioritizing the information and knowledge delivery through understanding the current context of a knowledge worker [42]. That is, a filtering mechanism provides the user only the information that is contextually related to the user's current task or project. The users are involved in creating and shaping their context of work via creating tags manually or automatically by their behaviours. The idea is based on the fact that users are generally busy with several different tasks during the day and they constantly have to switch and concentrate on a different one.

2.3. Solutions to find relevant information

To manage and store information sources in business organizations, it is a common practice to utilize document repository or knowledge management tools that facilitate sharing, reusing and managing information between employees. The problem with these tools is the difficulty of finding relevant information once it is shared in the system. The research area of information retrieval covers the approaches in order to successfully find the document or the information that is being searched for. In the 1960s information retrieval was defined as "a field concerned with the structure, analysis, organization, storage, searching and retrieval of information" [43]. Since then the area evolved into many different techniques and models in order to adapt to changing needs, such as exact match models [44], vector space models [45], and probabilistic approaches [18]. The latest approach is based on semantic approaches.

Storing and querying semi-structured data: In order to utilize heterogeneous and incomplete information data research and practice aimed at a semi structured format that is flexible and also appropriate for querying. Approaches for dealing with semi-structured data are XML and RDF and their query languages XPath and XQuery for XML and SPARQL for RDF. Especially XML is widely used in a variety of environments for managing and sharing loosely structured data that are represented in a hierarchical manner [44]. Lately RDF has gained the attention of researchers since it provides much more flexibility compared to XML by not enforcing a hierarchical structure, but supporting any kind of relations between data items.

Semantic Web technologies are the new generation of presenting and sharing data in various application areas. They started to be used in web platforms as well as tools that are in a way related to managing and providing important data [3]. The idea of a Semantic Web is to give information a well-defined representation so that it will be available in a more meaningful, structured and reusable way, which will enable humans and computers to work in cooperation to retrieve data from the Web [46].

In ontology-based Semantic Web applications, information is presented at a semantic level with ontologies, independent of the data structure and implementation, with a set of concepts and relationships between them [23]. This idea emerged from the need to enable some tasks to automatically understand the concepts in order to find relevant information, combine and share it with different resources. The representation of information with ontologies provides a common format between different systems and applications in order to share, understand and use knowledge [47]. This common format is standardized by W3C with the Web Ontology Language (OWL), Resource Description Framework (RDF), etc.

With the use of ontologies, a query is composed of entities from the ontology and their relations. This allows users to set the context of the input query. Moreover, usually in this kind of

data retrieval an external knowledge base is used to process the documents and the query. This knowledge base is used not only for text processing but also for solving the synonymy problem, as the synonyms of the words already exist in this database and are used during retrieval. Other than solving these two main problems in information retrieval, this method is also useful for extracting key knowledge from document sources. The query results are not only listed as documents, but also pure knowledge that is extracted from these documents. The information that is available in various documents and sources can be merged and brought to the user according to the query.

2.4. Ontologies in software engineering

Semantic Web technologies have been applied to different processes of software engineering in order to formalize information, improve access from different physical locations, improve universal information retrieval and allow checking and pairing different concepts and information [48], examples are ontologies for software processes [49], requirements [50], software architecture [51] and domains [52], and document ontologies [21].

All these ontologies are being used to improve software development. Their aim is to help software engineers to manage and understand large amounts of information in a shorter period of time. Although there are good examples of the usage of these ontologies, the area is still evolving and the usage of semantic technologies in software engineering will increase in the coming years with some improvements in Semantic Web technologies. The drawbacks for now are that constructing ontologies and implementing a Semantic Web enabled tool require a high investment of time. However, after the definition of ontologies, it is very flexible and easy to modify it according to the changing needs of an organization [37]. This also means that a dedicated person may be needed to maintain the semantic systems and their ontologies.

Although there are several studies that focus on developing ontologies related to software engineering processes, there are only few attempts to build an ontology that covers all software engineering knowledge. The most important among them is the work done to create a software engineering ontology based on the Software Engineering Body of Knowledge (SWEBOK) [53]. In SWEBOK a software engineering discipline is categorized into 10 knowledge areas. All these knowledge areas have their own processes and concepts. The proto-ontology, which was created based on SWEBOK, conceptualized all information in over 4000 concepts along with 400 relations and 1200 facts [54].

There are similar projects, such as Onto-SWEBOK, which are designed based on the 2004 Guide to the Software Engineering Body of Knowledge (SWEBOK) [55, 56]. However, none of them is released or publicly available because of unfinished projects due to the complexity, required time and human resources [55].

Another attempt to create a software engineering domain ontology is OntoGLOSE which is a light-weight global ontology [57]. This project uses the Glossary of Software Engineering Terminology published by the IEEE Computer Society [58]. The IEEE Glossary contains 1300 terms and their definitions that are related to the software engineering domain. The created ontology is composed of 1521 classes where each class has a unique meaning. Moreover, 329 relationships between classes were extracted using the semantic and linguistic analysis of the text in the glossary. As a result, OntoGLOSE is the only publicly available global ontology for the software engineering domain. The ontology does not have hierarchical classification; it rather forms a simple vocabulary and relationships among them that can be used for semantic annotation. The drawback of this ontology is that it is based on the IEEE Glossary, which was built in 1980 and updated in 2002, which means that it is out-to-date considering the amount of advances in the last 10 years. Moreover, the fact that it does not have any hierarchy, it is not the ideal way to structure information.

2.5. Tools to support ontology-based knowledge management systems

There are numerous tools that are developed in the vision of the Semantic Web. Below an

overview of the tools that might be related to developing a Semantic Knowledge Management System is presented.

The first step for a KM system is knowledge acquisition, and to acquire information from an unstructured text, several frameworks that can process plain text and extract concepts are used. GATE (General Architecture for Text Engineering)[3] is one of the most commonly used frameworks and has several plug-ins and integration capabilities [59]. It has many flexible language processing components that rely on finite state algorithms and the Java Annotation Patterns Engine (JAPE) language. It is widely used due to its precision for entity recognition and suitability for research as it is open source software. Moreover, it is commonly used in the semantic world because it offers full support for ontology integration. It has been utilized in ontology-based information extraction projects such as Multiflora, hTechSight and MIAKT [60].

IBM produced the UIMA[4] framework, which is an enterprise semantic search tool, but it does not provide full integration and support for ontologies [61]. Another tool is OpenNLP[5] from Apache, it supports many NLP tasks such as tokenization, segmentation, named entity recognition. However, it accomplishes these tasks via its built-in tools, not via any external ontology integration.

When it comes to Knowledge Representation, there are many tools to create, manage and edit ontologies. Protégé[6] is one of the most common open source ontology editors used by developers, researchers and corporations. It provides a user-friendly interface to build ontologies, knowledge-based tools and applications thanks to its support for plug-in extensions. GATE also has integration support for the Protégé tool.

Uren, et. al [28] provide a comprehensive work on the analysis of different annotation tools and frameworks, and offer a comparison of them.

[3]GATE: http://gate.ac.uk
[4]Framework UIMA: http://uima.apache.org
[5]Framework OpenNLP: http://opennlp.apache.org
[6]http://protege.stanford.edu

3. Method

This section illustrates the research method that was used based on the guidelines by Runeson and Höst [35]. In order collaborate with the industry, it was essential to first conduct a qualitative study to learn about the strengths and weaknesses of the solution (semantic knowledge management system), and obtain feedback from practitioners in the context. This also allowed the practitioners to learn about the semantic knowledge management system. The qualitative information could also be useful later to explain the reasons for quantitative results. In this sense, the study is of exploratory nature with a focus on qualitative data.

3.1. Research questions

In this study the following research questions were defined:

- RQ1 (Contribution 1): How to implement semantic knowledge management systems, and which challenges and impediments are observed?
- RQ2 (Contribution 2): How useful is the semantic knowledge management system perceived by software engineering practitioners?

The research process is conducted in three phases (see Fig. 1). The detailed phases are described in Section 3.3.

3.2. The case and unit of analysis

The case studied was a development site of Ericsson AB located in Sweden. The company is one of the leading telecommunication companies in the world and develops software in telecommunications and multimedia domain. The company products are used in more than 180 countries in the world. Currently the company has more than 100.000 employees.

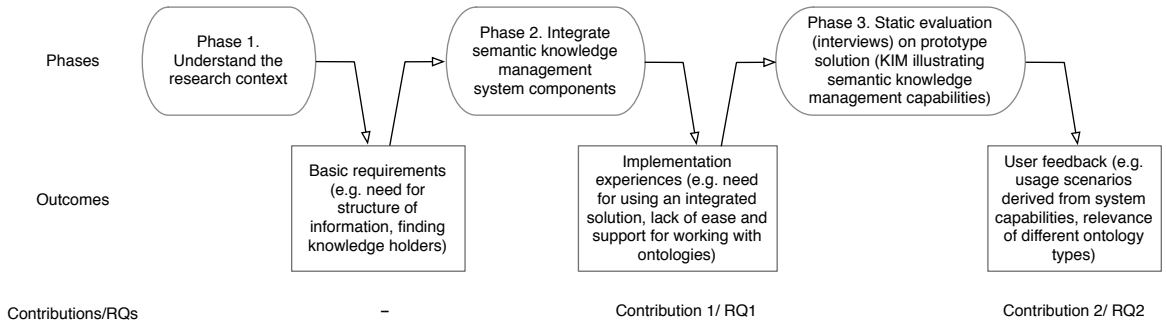


Figure 1. Research process

As far as units of analysis are concerned, internal knowledge management systems and the documentation they entail were defined as the unit of analysis. The case study design can be classified as a single case holistic design [62]. Ericsson uses a set of in-house knowledge management tools. These were platforms where everybody can share all sorts of information. They supported uploading documents and files; sharing blog posts and creating groups and discussion boards.

3.3. Data collection

The study was divided into three phases, understanding the challenges, implementation of the solution and its application in the company, and evaluation interviews.

3.3.1. Phase 1: Understanding the context of the organization

At the beginning, interviews were held with two key stakeholders from the organization to identify their needs and problems, they persons were the key contact persons and gatekeepers. The interviews were unstructured and were aimed to kick-off the project and achieve initial understanding. That is, the purpose was to get to know the company, project, problems in information retrieval, and responsible people. This was not a core part of the research (i.e. it is not reflected in the research questions), though contextual information was highlighted as essential when interpreting findings from case studies [35].

In order to elicit the requirements and issues, three separate meeting sessions were conducted. The first two meetings were held with the industrial contact who was supporting this study in the organization. He was a system level manager with over 20 years of experience. The meeting lasted an hour.

For the third meeting, two experienced software managers from Ericsson, who were responsible for innovation and had technical backgrounds in software engineering, were also invited. The goal was to see what was available in the literature and discuss the applicability of the desired solutions during the interview.

The following topics were discussed during these meeting:

- Introduction of the company and the responsible people to the student.
- Existing challenges related to finding information in the organization.
- Deficiencies of current internal collaboration tools.
- Requirements of a new solution.
- Possible usage scenarios about accessing relevant information.

The interviewer took notes during these interviews. Moreover, bi-weekly workshops were organized to discuss the findings, solution alternatives and status updates with the industrial contact. Hence, the data collected from the initial meetings was validated in these workshops. These meetings were important due to the possibility to obtain constant feedback from problem owners and also analyse the impact of the solution proposals on the company.

3.3.2. Phase 2: Development of a simple semantic knowledge management system

This phase required considerable time and effort in comparison to the other activities. After choosing the solution strategy in the initial interviews, an example system was created and applied to real world data to allow the participants to understand what the application of semantic technologies means in their context. The idea was not to implement a complete system that can replace the existing one, but rather to have a prototype which was sufficient to evaluate the usefulness of semantic systems in general.

Four different components to be supported by a complete semantic knowledge management system were defined and executed in this study. This section describes the components in general, while the details of the actual implementation are provided in Section 4.2.

Text processing (knowledge acquisition): The purpose of a semantic system was to extract knowledge from sets of unstructured information. Hence, the first step was to analyse and process these unstructured documents using the Natural Language Processing (NLP). The NLP technology has evolved to gain many capabilities in order to process the syntax and semantics of a text.

Ontology & knowledge base (knowledge representation): Ontology was one of the most important factors in information extraction as it provides conceptualization to the content of documents and was used for text processing. The ontology must be suitable to the contents of the information sources that are to be processed. Hence, there was a need to make a suitable ontology choice depending on the context of the domain.

Semantic annotation & ontology population (knowledge acquisition) & representation: When NLP tools parse the unstructured text, the entities found there should be annotated and mapped to the ontology. Therefore, the ontology could be populated with the extracted knowledge in the RDF or OWL format. This was the most significant step in information extraction as it was the phase where the relations between entities were defined. There were several platforms and ways to accomplish this step. Since this step was both depended on the NLP tool and the choice of ontology, it was crucial to choose a suitable system to integrate and work efficiently.

Semantic search (knowledge use): Once the ontology was populated with the instances and relations extracted from the text; the only step left was using a query language that was created for the Semantic Web in order to retrieve relevant information. A query engine needed to be chosen and should be supported by a graphical user interface. Users should be able to perform search with semantic capabilities, navigate between sources according to their semantic relations.

The details of the implementation and experiences made are presented in Section 4.2.

3.3.3. Phase 3: Evaluation interviews

The final evaluation and analysis was done by means of interviews with several company employees. This phase provided information about current challenges, obstacles about accessing information and possible improvements, suggestions and critique for the proposed new system. The system usefulness and users' experience with the system with semantic capabilities were evaluated. This time interviews were semi-structured. The prepared questions constituted a checklist of topics that should be covered during the interview.

Selection of interviewees: Knowledgeable practitioners should be chosen to conduct the interviews. Convenience sampling with diversity in mind was applied [63]. The interviews were conducted with employees with experience ranging from 3 to 25 years and with diverse roles, such as project manager, software architect, software developer, R&D specialist, solution architect.

As a result, eight employees were interviewed as can be seen in Table 1 below, which is believed to provide a sufficient amount of information to contribute to the literature and industry.

Interview guide: The interviews were related to the usage of internal collaboration tools of the organization, such as frequency of use,

Table 1. Interviewees

Role	Experience	Responsibilities
Project Manager	10 years in Ericsson 20+ in total	Project management, process improvement, process management
Software Architect	12 years in Ericsson 15+ in total	Software design, development, innovation
Senior Specialist R&D	20 years in Ericsson 25+ in total	Next generational rating and charging, information and business modeling
Solution Architect	7 years in Ericsson 10 years in total	Charging and mediation
Software Developer	2 years in Ericsson 3+ in total	Software customization center
Software Engineer	2 years in Ericsson 7 years in total	Software customization center
Solution Architect	19 years in Ericsson 20+ in total	Telecommunication services
Software Engineer	2 years in Ericsson 5 years in total	Proof of concept integration, Machine-to-machine applications

usage scenarios, satisfaction of the current version and suggested improvements. Later, the new semantic knowledge management system was presented to the users, and they were asked to explore the new system by using it. After they had gained an idea about the system, similar questions to the ones asked at the beginning were repeated and their opinions were collected and compared. The interview was structured as follows and the detailed guide is presented in Appendix A.:

- *Warm up:* First, the interviewer presented himself, the background of the project and the reason for making the interviews. Then the interviewee was asked general questions about their role, experience and current projects. This part of the interview was conducted mainly to build knowledge on the people and situation.
- *Information related to the usage of collaboration tools and problems:* It is important to know how and for which purposes people used the company's tools during their daily work. This part was devoted to figure out how often they used the current systems, how satisfied they were with the system (KIM, see Section 4.2.5) and what they would like to change in these tools. Basically more usage scenarios, requirements and problems with finding information were elicited.
- *Implicit knowledge:* It was important to learn how the employees gathered knowledge when they could not find what they looked for or when they were not satisfied with the findings they obtained. The authors tried to establish if they felt the need to talk to an expert and if so how they found out who the expert or responsible person was in that area, and so on. These questions are based on the data collected in the initial interviews.
- *Presentation of the prototype of the new system:* In this phase, an overview of the Semantic Web technologies was given and the information about the usage and goals of the Semantic Knowledge Management Systems were presented. Then the new system was presented as a prototype and the functionalities coming with the Semantic Web were explained. The interviewees were allowed to browse in the system documents for a while in order to make sure they were aware of the differences with the existing traditional knowledge management systems.
- *Satisfaction and evaluation of the proposed system:* The interviewees were asked to compare this system with the existing one. They were also requested to state whether they would use this system more often and if it would help them to make better decisions or reach implicit knowledge more easily. The point of the question was to capture the interviewees' attitude related to the evaluated system (KIM), as this was an important indicator for adoption and the possibility for a solution transfer from academia to industry. The actual decision quality could not be evaluated in this context.
- *Recommendations:* Finally the questions about possible different options for creating the Semantic Knowledge Management

Systems were presented and the interviewees were asked about preferences related to ontologies. Also, suggestions of improvements related to the proposed system were captured.

3.4. Data analysis

The qualitative data from the interviews was analysed using Thematic Coding Analysis (TAC). The authors followed the guidelines described by Robson [64] who describes TAC as a generic approach to analyse qualitative data, highlighting its flexibility, ease of application, and efficiency. The process was based on open coding and the identification of themes. The open coding was done manually on papers using color-coding, open codes belonging together were grouped during axial coding (referred to as themes).

3.5. Validity threats

We analysed the validity threats and mitigating factors in our case study following the descriptions given by Yin [62]:

Construct validity: Construct validity is concerned with the extent to which what was intended to be measured was actually measured [35].

- Selection of the Interviewees: The selection process was managed with the help of practitioners from the company. The selection process was a combination of diversity and convenience sampling. As far as convenience sampling is concerned, the selection was made based on the knowledge and availability of the employees. There is a risk that practitioners can choose people who support ideas similar to theirs. The usage of diversity sampling mitigated this threat by selecting employees with more diverse roles and experiences. At the end, the interviewee selection formed quite a diverse and potentially useful list of organization members.
- Reactive Bias: This one refers to the risk that the interviewees might be affected by the presence of the researcher and give biased answers that would influence the outcome of the study. This threat was partially reduced as a practitioner from the company was the gatekeeper who made the contact with interview candidates and helped build a trust relationship between the researcher and the interviewees.
- Correct Data: The correctness of the data aggregated by the interviews refers to the researcher's interpretation of what the interviewee actually said. To ensure this, all the interviews were recorded after taking permission from the interviewee so that any misunderstandings due to incomplete interview notes would not occur. Moreover, the interpretations of the interview transcriptions were sent back to the interviewees to obtain their validation feedback (member checking).
- Duration of the usage of the system: The practitioners used the system but only for a limited period of time. The practitioners know the existing system very well. Because the interviewees used the system themselves, they could, for example, understand its capability for different ways of searching (e.g. with regards to filtering specific entities that wpuld show only then and were unambiguously identified, see Section 4.2.5). Even though they did not have long-term experience, it was evident from their responses that they understood the concepts (and hence the opportunities) clearly, evidenced by the very informed feedback regarding Ontologies and Filtering (Sec. 4.2.3).

External validity: External validity is the ability to generalize the findings in a way that they will be interesting for other people representing other interest areas [35].

A single case company has been investigated. The results of single case studies in comparison to, for example, a survey have limited generalizability. However, the benefit are detailed explanations and a profound understanding of the situation that could be obtained. To reduce the effects the authors interviewed employees from different organizational parts of the company. Moreover, the context of the case study was described in detail (see Section 4.1), which allowed to map the findings to other large-scale organizations that are involved in software development.

Reliability: Reliability refers to the issue of finding the same results when the same study is replicated in the same setting [35]. The main threat to reliability are possible misunderstandings about the questions that were asked to interviewees, as they might have misunderstood the questions and hence provide answers different from the intended ones. An attempt was made to reduce this threat by keeping the questions as simple as possible. Open-ended questions were preferred so that the participants were encouraged to talk and express their opinions openly.

Internal validity: Internal validity concerns the validity of causal relations in explanatory case studies. It is related to the unconsidered factors that might have an impact on the relation [35]. The analysis of the usefulness of semantic knowledge management systems can be biased because of the employees' opinions about the existing systems. For instance, if the existing system had a search engine that is as powerful as the one Google applied to their documentation, the findings could potentially change.

4. Results

First the research context is presented, it is followed by the answers to the two research questions.

4.1. Phase 1: Context

A multinational large-scale organization like Ericsson has thousands of employees all around the world and hundreds of projects running in parallel. Considering the increasing amount of globally distributed projects in the software engineering domain, communication between team members is an essential part of software development. To increase the efficiency in communication, enterprises use knowledge management tools for enabling employees to find and share knowledge digitally. To share knowledge people used blogs, Wikis, discussion boards, project contents and documents. Since all Ericsson employees, i.e. more than 100.000 people, use these tools; there are large amounts of documents. All these documents and information are not stored in a structured way and, hence, it is necessary to find ways of managing this large volume of unstructured data. It is imperative to investigate how to overcome these problems. All the interviewees mentioned that the existing search facility does not satisfy existing needs and so a more intelligent solution should be found. In particular semantic knowledge management and ontologies allow to bring structure to the information stored, which was one of the motivations for the company to participate in the study.

The following challenges and needs of the organization were raised during the initial meetings to understand the context:

- The practitioners defined usage scenarios that are common, in particular active search based on queries, passive search, analysis of contributors (users), and the analysis of trends. Overall, the practitioners identified scenarios that are common and well understood in the knowledge management community.
- Structure of information was a common issue, which is not specific to the company. Bringing structure to information is well supported by ontologies, making them interesting for the company. Performance issues and formatting were specific for the company and could be easily improved.
- The search engine used at the company is perceived as a poor quality one.
- Filtering of search results and complicated structures have been highlighted, which is also a good motivation for annotating documents and mapping them to an ontology in the context of a semantic knowledge management system.
- A challenge was also finding an expert, which is recognized as a key challenge in literature, too [65].

In summary, the conclusion was that the search should be improved, and that semantic knowledge management systems could be a potentially useful solution.

4.2. Phase 2: Development of a Simple Semantic Knowledge Management System (RQ1)

This is the phase where it was necessary to make a comprehensive research and spend time and

effort on the development of a new knowledge management system, which took a total of four person months. One of the reasons for the effort needed was the absence of information about how to implement a semantic knowledge management system in the literature. Although the final solutions were presented in some studies, the way to implement them was barely mentioned. For this reason, this section will illustrate the steps to accomplish this goal and the results gathered during the process. An important detail about the following two sections is that, they are not necessarily sequential processes; ontology building was performed simultaneously when the development attempt was made.

It is important to point out that the attempt to build the semantic knowledge management system based on components was not successful for the above mentioned reason. The best working solution was to utilize an integrated solution (KIM), which is described in Section 4.2.5. As KIM is easier to use, a transfer to the software industry for knowledge management purposes is more likely. Thus, KIM is used in the subsequent steps of the study (i.e. Phase 3). The principle architecture of semantic knowledge management systems and how it relates to KIM is shown in Figure 2. The details of the KIM platform are further elaborated in Popov et al. [27].

4.2.1. Ontology building

The first step to build an ontology is determining the domain and the scope [66]. In this case, the domain are all kinds of knowledge that can be shared in Ericsson software projects. That is, the ontology should cover aspects from generic software engineering domain to the company domain. The latter can be considered as the projects, characteristics of projects, employees and terms related to the telecommunication domain. However, in the scope of this work, the focus will be more on the concepts that are directly related to software engineering. The specific terminology of the company will be left for future research. The usage purpose of this ontology is to categorize all the necessary information about software engineering that might be shared in collaboration tools. Considering the usage scenarios that are defined in the previous section, one can say that the ontology should only be sufficient to cover the topics that organizational members can possibly share or mention. Hence, the ontology should provide answers to such questions as people's interests, expertise, projects, locations of projects and people.

The second step in building an ontology is considering reusing existing ontologies instead of creating a new one [66].

There have been several studies about building ontologies in software engineering. Most of these attempts focused on specific phases of software engineering, such as requirements, architecture, implementation, testing, maintenance [20–22, 50, 67]. However, there are not many projects that try to develop ontologies that fully conceptualize all the knowledge in the field of software engineering. The major efforts to achieve this goal are aimed to adopt the SWEBOK Guide as a formal ontology. Such an ontology would be a good choice for the scope of this research as it would cover all the content and terminology in the software engineering domain. Unfortunately these attempts have not yet been successful or completed due to its complexity and required effort [54–56].

As a result, a decision was made to work with the only successfully released global ontology OntoGLOSE, which is based on the IEEE's global terminology for software engineering [57]. Although there are certain drawbacks of this ontology, such as the lack of coverage and the fact that it is outdated and primitive; utilizing this lightweight ontology would still be sufficient for the scope of this study to reach the current research goals.

4.2.2. Text processing

For processing an unstructured text, it was decided to use GATE due to its common usage in semantic web research and support for ontology based information extraction. GATE comes with an information extraction system called ANNIE (A Nearly-New Information Extraction System). Using ANNIE's components such as tokenizer, gazetteer and sentence splitter; one can extract

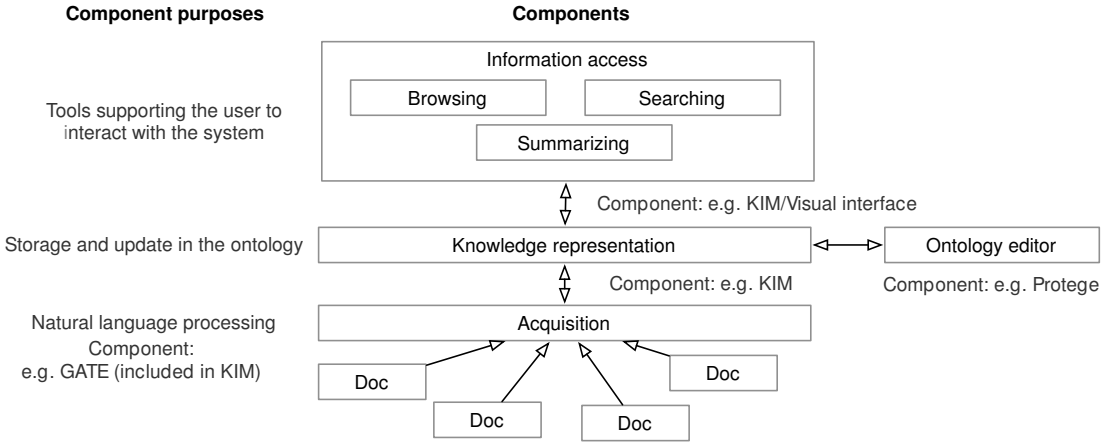


Figure 2. Semantic Knowledge Management Architecture

generic information from the corpora of the unstructured text. GATE can find the names of well-known organizations, names of people, locations, numbers, etc.

When GATE and ANNIE were applied to some documents from the knowledge management system of the company, it could be observed that the recognized entities were not enough to cover the content and the context of the technical documents that were used, such as domain specific terms that were relevant for the practitioners, but not highlighted.

Hence, to extract the information related to software engineering, a suitable ontology should be integrated as a language resource and the necessary changes to the processing resources of GATE should be reflected.

4.2.3. Ontology and knowledge base

After understanding how to use GATE, the next step was investigating how ontologies can be involved in text processing. The decision was building a simple ontology in order to have an initial idea about the usage of ontologies.

In the process of building and managing ontologies, Protégé was selected as an ontology editor for several reasons. First of all, Protégé is an open source research project which is extensively used in the academic world. Moreover, the authors had previous experience in using this tool in another academic project. Finally and possibly most importantly, GATE and Protégé support integration for each other and support many other tools and extensions.

For this initial phase, a very simple ontology that already covers some of the content of the document was built and used for testing text processing and annotation. Later Protégé was used to manage the existing ontologies as described in the previous section.

4.2.4. Semantic annotation and ontology population

A fully automatic semantic annotation tool is needed to apply it and evaluate directly on the corpus of the organization's knowledge management systems. Manual annotation tools require user intervention, so their usefulness cannot be directly evaluated without manually populating them with information.

Finally, the decision was to use GATE also for semantic annotation as it supports the automatic annotation of documents. Therefore, it was used to make an initial attempt to annotate a company document with the built ontology. At the end, GATE was used for NLP and semantic annotation and Protégé was used for building ontology.

After exploring the tool for a while and gaining the understanding of how it worked, it became clear that adapting the processing resources of GATE was not such an easy task and might require a lot of effort. First of all, building a knowledge base, creating instances for each entity of

the ontology, which would be sufficient for evaluating the system during the case study could not be done manually within the time and resources provided by the company and the research project. Choosing an external knowledge base and integrating it would also mean a need for a substantial amount of time,. Moreover, even though the knowledge base can be integrated, the GATE annotation system should be modified so that it can recognize and instantiate the relations between entities. Based on the tutorials and documentation of the framework, this requires advanced NLP expertise and a considerable amount of research and effort.

In addition, after this step a query engine with a graphical user interface needs to be implemented, which would require a significant amount of time as well. Considering the time constraint, a decision was made to make a mind switch and look for alternative solutions. The authors looked for integrated platforms that use GATE and also provide semantic search facilities with ontologies.

4.2.5. Integrated semantic knowledge management platform (KIM – Knowledge and Information Framework)

The decision to use the KIM platform (see Section 2) was made because it met the requirements of this study and the defined usage scenarios. Some reasons why the other platforms could not be used encompassed the fact that OntoShare is not available online, Semantic Wiki and ACTIVE cannot be applied to existing knowledge management systems, they need to be built as a new system. Moreover, they do not satisfy the initial requirements for solving search problems. KIM supports the fully-automatic semantic annotation of documents and comes with an upper-level ontology and a semantic search engine. KIM is based on GATE for NLP purposes. It comes with an ontology named PROTON[7] that covers the most general concepts, such as named entities (people, locations, organizations) and concrete domains (numbers, dates, etc.). However, a more specific ontology can be integrated with KIM according to the needs of the domain. The stated requirements were analysed and compared with what KIM can offer and, in consequence, the following results were achieved:

- KIM's general ontology covers most of the aspects defined in the scope of the ontology for the purpose of this study. There is no need for numerous changes in the ontology design such as classes and relations. It is possible to integrate the OntoGLOSE domain ontology and this will enable KIM to recognize domain specific concepts. There is no need for very specific relations between classes as our usage scenarios are only based on extracting who is talking about what topic, either. As long as the topic is recognized, it would be sufficient to satisfy the specified requirements.
- If the domain ontology is not enough to cover all the aspects, as it does not have any concepts developed in the last 10 years and many other concepts about the company domain, the KIM knowledge base can be extended with an external knowledge base. For instance, KIM supports integrating KIM with DBpedia[8] which is a structured knowledge base containing all Wikipedia entries. Considering the fact that Wikipedia contains all the terminology that we need for software engineering as well as the telecommunications domain, integrating DBpedia would be a convenient solution.
- KIM provides "Boolean Search" which is a keyword-based search and corresponds to "Active Search" in defined usage scenarios. Moreover, it provides "Structure" and "Pattern" search in order to search for the extracted relations which can be used for the "Finding the Tribe" scenario. "Facet search", which is a relational filtering mechanism, can also be used for the same scenario. "Timeline" search, which shows the popularity of selected entities over a period of time, can be used for the "Trends" scenario defined by the authors. On the other hand, KIM also provides navigation between documents according to their relations, which enables "Passive Search". The KIM search frame and the "Structure" search menu can be seen in

[7]PROTON: http://proton.semanticweb.org
[8]DBpedia: http://dbpedia.org/

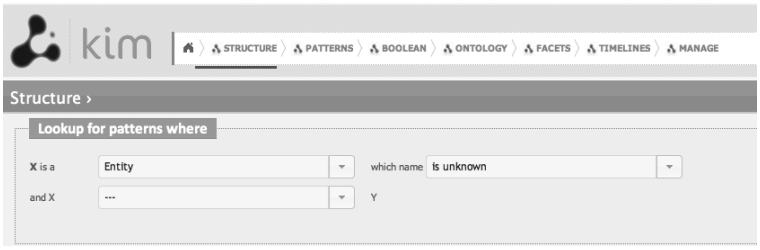


Figure 3. Structure Search from KIM

Figure 4. PROTON ontology

Figure 3. KIM has the capability of detecting persons in the text through the basic PROTON ontology. That is, if the same person for example always appears in blog posts, discussions or other documents in relation to a specific product, then this person could be an interesting contact. This can be captured as a semantic query can find relations between entities, such as people and topics, hence supporting the "Tribe scenario".

First of all, an attempt was made to integrate DBpedia with KIM. To be able to use the DBpedia instances, it was necessary to to integrate the DBpedia ontology with PROTON, which is the generic ontology of KIM. However, the whole DBpedia ontology be mapped to PROTON as it would cause too much complexity. Therefore, Person, the Organization and Abstract classes of DBpedia were taken and mapped to PROTON, so that the names of all well-known people, organizations and also abstract topics which contain the software engineering related topics are included. Figure 4 represents a part of the PROTON ontology and its Person and Organization classes.

However, due to poor documentation and the lack of available external support and expertise, it was not possible to successfully integrate DBpedia to the KIM knowledge base. Integrating DBpedia consists of many steps, such as mapping of ontologies, adding statements for each entity in DBpedia, setting labels of each entity, setting up gazetteers for each newly added class, adding Jape transducers and so on. Hence, the documentation for such complex tasks should be clear and detailed, so that developers with no extensive experience can also accomplish them.

Therefore, it was decided to integrate the software engineering domain ontology OntoGLOSE. Although it did not satisfy all our needs, it was be a good starting point for a further study to modify and extend its coverage.

After integrating this domain ontology, it was established that the system still did not to recognize the entities in this ontology. As a backup solution, a manual integration of this ontology to the actual PROTON ontology was conducted via Protégé Ontology Editor. Since OntoGLOSE does not have any hierarchy, it was easy to manually copy its classes to the other ontology. However, the relations were neglected as they were not interesting for this context. However, even after these steps were taken, KIM still did not manage to recognize these terms. The company was consulted by e-mail, but due to the long delays in getting a reply from the support team, there were only two e-mail exchanges with them, which was not enough to fix the problems.

Therefore, the system was evaluated during the interviews as it was. The discussion board pages from one of Ericsson collaboration tools were downloaded manually and loaded to KIM as a corpus. There was no quantitative measure, though too much information was in the system to easily search it. Thus, the corpus was sufficient to evaluate the usefulness from an end-user perspective during the interviews as they could experience the main concepts of the semantic knowledge management system.

The key findings for RQ1 are presented below.

Key findings and observations for RQ1:

i) It is time intensive to build a semantic knowledge management system, in particular setting up the ontology is a great challenge which required the majority of the effort.
ii) Rather than integrating different parts of a semantic knowledge management system, it is recommended to use an integrated platform as it is easier and hence more likely transferable to industry. Thus, KIM was used in Phase 3 of the study.
iii) Different types of searches (in particular pattern making use of the ontology) are possible with KIM, hence making explicit use of ontologies.
iv) KIM does not allow to easily integrate ontologies other than PROTON, which is a limitation. Beyond that KIM is easy to use.

4.3. Phase 3: Evaluation interviews (RQ2)

In Phase 3 the reflections of the practitioners on the usefulness of KIM, the ontology and filtering as well as possible improvements to the knowledge-based system, are discussed.

4.3.1. Usefulness of KIM

All of the interviews confirmed that the overall approach that comes with the semantic systems seems very useful. Although they all remarked that their current search engine was totally incapable and the proposed one (the new one?) cannot even be compared to the existing one, they pointed out some strong points of the semantic search.

Finding documents and faceted search: All of them found it useful to search for documents with their relation to people, topics and authors. However, they suggested different ontology alternatives, which will be discussed in the "Ontology and Filtering" section below.

Two interviewees found "Faceted search" the most useful, as it starts broader and narrows down based on the results of added filters. One of them stated that "I like the idea of refining the search. Start broader and then based on the result, narrow it down. That's a good way to search. Because that's the way you search normally, going from broader to specific." One interviewee indicated that being able to see all the extracted information without even making a query is very useful because you can see beforehand if it is worth your time looking into the database.

Finding eople and their position, roles and locations: Most of the interviewees (6 of 7) also agreed on the usefulness of this system about finding people, which was previously defined in this study as "Finding the Tribe" in usage scenarios. One subject mentioned that they did not need this functionality because they knew everybody he needed . Others stated that finding experts and knowledgeable people was quite a common scenario in Ericsson as there are experts in almost every area and their knowledge

is indispensable. One of them added that, "Finding the right person was a common practice in Ericsson. It is a large organization. Not everyone knows everything but you can find an expert in almost every area. However, sometimes you don't know who they are. You should be very active in forums, etc., but it needs spending time on them regularly. So this facet search is very, very useful." They all agreed that the correct recognition of software engineering and telecommunication terms by the NLP tool is crucial for the success of this search engine. Two interviewees indicated that extracting organizational information about people's position, roles and locations would not be necessary or useful since this information is actually stored somewhere in the company database. However, they would like to integrate this database, which is not directly accessible for employees, to this semantic system so that they could utilize organizational data while searching.

Extracting statistical data and decision making: Another point that the interviewees mentioned was the statistical data that could be gathered by means of this new system, which is similar to the "Trends" in usage scenarios. By analysing what people talked about, a significant amount of hidden data might be collected. For instance, people's skills and interests can be identified by processing the entries they are involved in. Furthermore, a summary of what people communicate about can be extracted with this system to make an organizational decision. Another example given by an interview subject was as follows: "If we have a lot of people working with GUI in a unit, or the majority of graphical people in Ericsson work in this city, maybe we should set up a centre there. This will mean that the statistics that we need are available directly there. Even if people don't update their profiles, they write documents so they will be recognized anyway." Another interviewee suggested that this kind of information about trends and statistics could be useful for sales people who go to customers. The connection to software engineering is not immediately evident. Though, in the context of continuous integration, customer relations in the organization are tightly coupled with software development, e.g. to enable continuous releases. Also, information from and to sales/customers is essential and becomes a part of guiding development and testing effort, as well as giving input to requirements engineering. In particular, from the point of view lean software development perspective, it is important to take an end to end perspective, from inception of an idea to sales and deployment.

4.3.2. Ontology and filtering

Practitioners were generally excited about the use of ontologies and making structural searches with respect to the ontology. However, none of them was directly interested in seeing a software engineering ontology with all the practices in the domain. They stated that their search scenarios are more about terms in the Telecom domain.

Ontology complexity and structure: A practitioner mentioned his concerns about the use of ontologies as an ontology can become quite big and have a lot of branches, which makes it too complex. Repeated breaking down the information to branches might make people lose track and become confused. He stated that "Although the usage of taxonomies is good for a human brain to understand, people might easily get lost in it if it gets too large." Hence, creating a complete ontology that has all the information structured in a certain domain would probably be too enigmatic and cause information overload problems. Another interviewee foresaw this and suggested gaining the ability to search in the ontology as well. This can prevent people from getting lost in the branches of the ontological structure.

Another point the practitioner mentioned was the fact that there was no complete tree structure. This interviewee suggested keeping the ontology very general and focusing on the tagging system.

Usefulness of the SWEBOK ontology: When it comes to the choice of ontologies, interviewees were asked if they would like the see knowledge areas based on SWEBOK in the ontology structure so that they could use them to extract and filter information. However, all of them stated that they did not really need

that kind of queries and one subject stated that these knowledge areas and lifecycle phases were not very clear when you used agile development. They declared their own choice of ontology would be useful for them.

Document type ontology: Document types and domains were the most desired ontologies by the interviewees. Three subjects specified that they would like to see the document types in the ontology so that they could filter the documents according to type. All the interviewees were asked to discuss their usage scenarios for these collaboration tools and the type of documents they dealt with. For the document types they gave the following examples: product description documents, project planning documents (requirements, user stories), design documents, business process modelling documents, architectural documents, release packages, CPI (customer product information) documents, operational documents, test reports, proposal, pre-sales and after sales documents, installation documents, solution documents, interface description documents, user guides and so on.

One interviewee mentioned problems related to the document type by stating that "The problem with document types is that there is no common structure about where to place these documents in the project repository. It can be anywhere." Hence, the partcipants could not easily find a specific document for a certain project or product. One interviewee denoted that if the semantic system could recognize the type of the document automatically by processing the content of the document, it would be a benefit for them.

Telecom domain ontology: Another common suggestion was a domain ontology based on telecom operations and services. Four interviewees mentioned that when they searched for a term, the results came from all different domains that were not interesting for them. When they were asked about what exactly they meant when they sais domain, one interviewee only stated that he would like to see only the results from the network (technical) domain or from the business domain. The other three participants were slightly more specific and they gave the following examples: Operation Support Systems (OSS), Business Support System (BSS), Charging, Mediation, Service Delivery Platform, Customer Relationship Management (CRM), etc.

They suggested using eTOM[9] (Enhanced Telecom Operations Map) which is a guidebook that defines the most common standards for business processes in the telecommunications industry.

The interview subjects indicated that they would like to have a combination of the domain, the document type and the organizational structure of the company when they create a search query. The organizational structure refers to the existing structure of the tools, which is based on location, region, unit, project, etc.

Organization-specific ontology: Another subject proposed the Ericsson project management framework PROPS-C as an alternative to the classical lifecycles defined in SWEBOK. This framework includes the business readiness, sales and project management processes. They are all composed of such phases as analysis, planning, monitoring, execution, contract management, etc. The interviewee suggested searching for documents according to these defined phases.

The same subject proposed to have the Ericsson Product Catalogue domain in the ontology. He said that "There are products and services such as network optimization and project management. When I make a project somewhat related to a product in the catalogue domain, I do not enter this project as a product because it is only a small part of it. Normally I put this document as a project under my unit. If I don't advertise this as a knowledge object or something, nobody can find this project. If I can relate this project to some place in the product catalogue, then it will increase its possibility to be found." This is important because other people might have similar projects that are related to only some part of the main products, however, the information about these projects ia lost in local repositories. Hence, relations between projects and the products from the catalogue can be useful for finding documents.

[9]eTOM: http://www.tmforum.org/BestPracticesStandards/BusinessProcessFramework/6637/Home.html

4.3.3. Improvements for the knowledge-based system

As far as the proposed semantic system is concerned, interviewees mainly made comments about the content of the ontology as it shapes the search mechanism. However, they mentioned some improvements that can be applied in the system.

Search mechanisms: First of all, one interviewee stated that they do not want to be locked into a set of predefined queries when making a structured search based on the entities and their relations in the ontology. He would prefer to write a search sentence; the system should semantically process it and, if it matches any of the relations in the ontology, then results should be retrieved based on that, otherwise it should perform a standard search.

Another suggestion was the ability to search for entities that do not satisfy the relation specified in the search pattern. For instance, searching for people who talk or do not talk about a certain topic should be available. He explained his concern by stating that "For example if competitors in our knowledge base haven't talked about something, it means that we don't have any understanding about what they are doing. Because they must talk about it."

Moreover, three interviewees suggested jumping to similar documents based on the overall content of the document. The existing system only allows jumping between documents based on a single annotation inside the document. This suggestion was identified as "Passive Search" at the usage scenarios in the beginning of the case study.

Tagging: All the interviewees at some point mentioned tags and they pointed out the importance of an intelligent tagging system. They indicated that tags are very useful for understanding the context and content of a document and a search engine should consider tags in a smart way in the search algorithm. However, they all agreed that tags in the current system were not used efficiently at all. One interviewee stated that people did not know the purpose of tags so they just wrote something or left it empty. Another interviewee mentioned that people do not have the patience to write proper tags so they do not pay much attention. He says people should not be forced to tag.

Three of the subjects proposed to have a closed solution for tags. One interviewee said that "In the case of an open-ended solution, someone will eventually tag in a different way and it will be problematic." The current system has a tag library and people can choose tags from there but they can also add any tag to the library without any supervision and control. The interviewee found this system messy and not usable.

However, the interviewees opposed to the introduction of a fully automatic solution. That is, they want to be able to modify the tags of documents even if they are not the authors and add new tags to the tag library. However, the tag library should be very wide and well controlled. Hence, they prefer a semi-automatic tagging system. This also applies to the semantic system proposed as the annotation and then the tagging is fully automatic. Moreover, one interviewee suggested binding tags with entities in the ontology which are able to search according to those tags. Currently the semantic system uses the most frequent annotations as tags but it is not possible to modify them. Another interviewee suggested having descriptions for tags. This is possible when the annotations are used as tags because recognized entities already have their descriptions.

Results presentation: Furthermore, some participants suggested improvements in the representation of the results. For example, one of the subjects wanted to see the tags or the summary of the document directly in the search results so that it can help them to choose the document with the right context. Another practitioner proposed to have results collapsed according to the ranking and organizational structure. In this way one can have traceable trees based on location, product, etc.

The key findings for research question RQ2 are stated below.

Key findings and observations for RQ2:

i) The ontologies related to software engineering were not of the main interest to practitioners.

They were more interested in domain-specific ontologies and document ontologies (recognizing a document type).

ii) The practitioners were positive about the different search options in KIM, in particular the Facet search and the Structural search. Being able to see extracted information without making a query is of great interest, however, it is not provided by traditional search tools. This also facilitates easy filtering, which was important to them.

iii) It is important to have simple ontologies to be still understandable.

iv) There should be a possibility to filter a search query by the domain, document type, and organizational structure.

v) The costs of implementation, migration, and maintenance have been raised as an important factor.

vi) In summary, the interview subjects denoted diverse opinions about the use of ontologies and what type of ontology they would like to see. However, the domain and documentation seem to be most dominant ones.

5. Conclusion

In this work, the main contribution was the analysis of the usefulness and applicability of ontology-based semantic information retrieval technologies in knowledge management systems in the context of software engineering in large-scale organizations. To perform this analysis from all perspectives, we identified the existing problems, available technology, useful aspects and challenges that the organizations should be aware of. The problems are related to the search engine and the structure of the existing tools, the technology is able to process documents to extract the knowledge inside, useful aspects are related to filtering out irrelevant documents and extracting people's skills and interests, and the challenge is the necessary effort to satisfy all the needs. The research questions asked can be answered as follows.

RQ1: How to implement semantic knowledge management systems? First individual components were implemented and an attempt was made to integrate them. This was a considerable effort, and the use of an already integrated solution (here KIM) was preferred. Still, the difficulty of integrating and updating new ontologies was high. It was found that practitioners need tailored ontologies, which is a hindrance for technology transfer. In general, the KIM system should reuse existing components (e.g. GATE) and ontologies as much as possible. However, the difficulty was to actually work and integrate the components. Even with the integrated solution, it was difficult to add and modify ontologies.

RQ2: How useful are semantic knowledge management systems in finding relevant knowledge in software engineering? The key part of a semantic knowledge management system is the ontology to be used, as the most beneficial structure has to be found. So far, we could not find any completed and released software engineering ontology that covers all the knowledge in the domain. Yet, the case study revealed that this was not necessarily needed. It was found that the practitioners mostly need a document ontology so that they can filter documents by their type and content.

Moreover, when it comes to reusing knowledge, it was observed that the business domain of the organization was equally if not more important, the practitioners indicated that the information they reuse or search is often related to domain specific knowledge, solutions, products, business processes, etc. Hence, the ontology should cover these aspects so that they can filter the documents accordingly. They proposed ontologies that cover business process frameworks for telecommunications (eTOM), organizational structure of the corporation, project management framework of the organization (PROPS-C) and the product catalogue of the company.

Overall, when looking at the initial requirements one may reason on their fulfilment.

– Structure of information: The need to structure information and making people aware of this structure was highlighted as very important. A means to do this are ontologies. Given the difficulty of updating and adding

new ontologies, the requirement has only been partially fulfilled.
– Finding experts: This also requires the update of the ontology incorporating organization-specific roles and terminology. Hence, only with an easy updating method, this would be achieved.

Future work: A replication the case study can be conducted in another large-scale company that operates in a domain other than telecommunications. The comparison of the two would yield important results about interviewees' ontology choice. It is essential to see if their main ontology choice is also based on the business domain of the corporation. To generalize the needs of software engineers about ontologies, it is necessary to conduct several case studies. On the other hand, another company in the telecommunications domain should also be analysed in order to remove the defined external validity threats. Also experimentation is needed. That is, in future work, the actual time to find information should be measured and also the quality of the decisions should be evaluated. This study may help in formulating research propositions as well as providing explanations for quantitative findings.

Acknowledgments

The work was partially supported by a research grant for the ORION project (reference number 20140218) from The Knowledge Foundation in Sweden.

References

[1] J.L. Krein, P. Wagstrom, S.M. Sutton Jr, C. Williams, and C.D. Knutson, "The problem of private information in large software organizations," in *Proceedings of the 2011 International Conference on Software and Systems Process*. ACM, 2011, pp. 218–222.

[2] E. Carmel and R. Agarwal, "Tactical approaches for alleviating distance in global software development," *IEEE Software*, Vol. 18, No. 2, 2001, pp. 22–29.

[3] J. Grudin, "Enterprise knowledge management and emerging technologies," in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, Vol. 3. IEEE, 2006, pp. 57a–57a.

[4] M. Alavi and D.E. Leidner, "Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues," *MIS quarterly*, 2001, pp. 107–136.

[5] C.Y. Yang and S.J. Wu, "Semantic web information retrieval based on the Wordnet." *International Journal of Digital Content Technology & its Applications*, Vol. 6, No. 6, 2012.

[6] J. Mustafa, S. Khan, and K. Latif, "Ontology based semantic information retrieval," in *4th International IEEE Conference Intelligent Systems*, Vol. 3. IEEE, 2008, pp. 22–14.

[7] W. Wei, P.M. Barnaghi, and A. Bargiela, "Semantic-enhanced information search and retrieval," in *Sixth International Conference on Advanced Language Processing and Web Information Technology*. IEEE, 2007, pp. 218–223.

[8] A. Edmunds and A. Morris, "The problem of information overload in business organisations: A review of the literature," *International journal of information management*, Vol. 20, No. 1, 2000, pp. 17–28.

[9] M.J. Eppler and J. Mengis, "The concept of information overload: A review of literature from organization science, accounting, marketing, MIS, and related disciplines," *The information society*, Vol. 20, No. 5, 2004, pp. 325–344.

[10] O.E. Klapp, *Overload and boredom: Essays on the quality of life in the information society*. Greenwood Publishing Group Inc., 1986.

[11] J. Feather, *The information society: A study of continuity and change*. London: Facet Publishing, 2004.

[12] H. Butcher, *Meeting managers' information needs*. London: ASLIB/IMI, 1998.

[13] R. Guha, R. McCool, and E. Miller, "Semantic search," in *Proceedings of the 12th international conference on World Wide Web*. ACM, 2003, pp. 700–709.

[14] N.J. Belkin and W.B. Croft, "Information filtering and information retrieval: Two sides of the same coin?" *Communications of the ACM*, Vol. 35, No. 12, 1992, pp. 29–38.

[15] C.J.V. Rijsbergen, *Information Retrieval*, 2nd ed. Newton, MA, USA: Butterworth-Heinemann, 1979.

[16] G. Salton, A. Wong, and C.S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, Vol. 18, No. 11, 1975, pp. 613–620.

[17] P. Warren, "Building semantic applications with SEKT," in *Integration of Knowledge, Semantics and Digital Media Technology, 2005. EWIMT 2005. The 2nd European Workshop on the (Ref. No. 2005/11099)*. IET, 2005, pp. 429–436.

[18] C.D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press, 2008, Vol. 1.

[19] A.K. Jain, M.N. Murty, and P.J. Flynn, "Data clustering: A review," *ACM computing surveys (CSUR)*, Vol. 31, No. 3, 1999, pp. 264–323.

[20] I.N. Athanasiadis, F. Villa, and A.E. Rizzoli, "Enabling knowledge-based software engineering through semantic-object-relational mappings," in *Proceedings of the 3rd International Workshop on Semantic Web Enabled Software Engineering*, 2007.

[21] R. Witte, Y. Zhang, and J. Rilling, "Empowering software maintainers with semantic web technologies," in *The Semantic Web: Research and Applications*. Springer, 2007, pp. 37–52.

[22] C. Kiefer, A. Bernstein, and J. Tappolet, "Analyzing software with iSPARQL," in *Proceedings of the 3rd ESWC International Workshop on Semantic Web Enabled Software Engineering (SWESE)*, 2007.

[23] Y. Zhao, J. Dong, and T. Peng, "Ontology classification for semantic-web-based software engineering," *IEEE Transactions on Services Computing*, Vol. 2, No. 4, 2009, pp. 303–317.

[24] B. Decker, E. Ras, J. Rech, B. Klein, and C. Hoecht, "Self-organized reuse of software engineering knowledge supported by semantic wikis," in *Proceedings of the Workshop on Semantic Web Enabled Software Engineering (SWESE)*, 2005.

[25] E. Simperl, I. Thurlow, P. Warren, F. Dengler, J. Davies, M. Grobelnik, D. Mladenic, J.M. Gomez-Perez, and C.R. Moreno, "Overcoming information overload in the enterprise: The active approach," *IEEE Internet Computing*, Vol. 14, No. 6, 2010, pp. 39–46.

[26] D. Hyland-Wood, D. Carrington, and S. Kaplan, "Toward a software maintenance methodology using semantic web techniques," in *Second International IEEE Workshop on Software Evolvability*. IEEE, 2006, pp. 23–30.

[27] B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, and A. Kirilov, "KIM – A semantic platform for information extraction and retrieval," *Natural language engineering*, Vol. 10, No. 3-4, 2004, pp. 375–392.

[28] V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. Ciravegna, "Semantic annotation for knowledge management: Requirements and a survey of the state of the art," *Web Semantics: science, services and agents on the World Wide Web*, Vol. 4, No. 1, 2006, pp. 14–28.

[29] T.R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?" *International Journal of Human-Computer Studies*, Vol. 43, No. 5, 1995, pp. 907–928.

[30] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig *et al.*, "Gene ontology: Tool for the unification of biology," *Nature genetics*, Vol. 25, No. 1, 2000, pp. 25–29.

[31] A. Kanso and D. Monette, "Foundations for long-term collaborative research," in *Proceedings of the 2014 ACM International Workshop on Long-term Industrial Collaboration on Software Engineering (WISE 2014), Vasteras, Sweden, September 16, 2014*, 2014, pp. 43–48.

[32] V. Garousi, K. Petersen, and B. Özkan, "Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review," *Information & Software Technology*, Vol. 79, 2016, pp. 106–127.

[33] A. Arcuri, "An experience report on applying software testing academic results in industry: We need usable automated test generation," *Empirical Software Engineering, in print*, 2017, pp. 1–23.

[34] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, "A model for technology transfer in practice," *IEEE Software*, Vol. 23, No. 6, 2006, pp. 88–95.

[35] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical software engineering*, Vol. 14, No. 2, 2009, pp. 131–164.

[36] R.J. Thierauf, *Knowledge management systems for business*. Greenwood Publishing Group, 1999.

[37] J. Davies, D. Fensel, and F. Van Harmelen, *Towards the semantic web: Ontology-driven knowledge management*. John Wiley & Sons, 2003.

[38] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff, "Semantic annotation, indexing, and retrieval," *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 2, No. 1, 2004, pp. 49–79.

[39] S. Schaffert, F. Bry, J. Baumeister, and M. Kiesel, "Semantic wikis," *IEEE Software*, Vol. 25, No. 4, 2008, pp. 8–11.

[40] E. Oren, M. Völkel, J.G. Breslin, and S. Decker, "Semantic wikis for personal knowledge manage-

ment," in *Database and Expert Systems Applications*. Springer, 2006, pp. 509–518.

[41] P. Warren, J.M. Gómez-Pérez, and C.R. Moreno, "ACTIVE – enabling the knowledge-powered enterprise," in *International Semantic Web Conference (Posters & Demos)*, 2008.

[42] V. Ermolayev, C.R. Moreno, M. Tilly, E. Jentzsch, J.M. Gomez-Perez, and W.E. Matzke, "A context model for knowledge workers," in *Proceedings of the Second Workshop on Context, Information and Ontologies*, V. Ermolayev, J.M. Gomez-Perez, P. Haase, and P. Warren, Eds., 2010.

[43] R. Baeza-Yates, B. Ribeiro-Neto *et al.*, *Modern information retrieval*. ACM press New York, 1999, Vol. 463.

[44] T. Calders, G.H. Fletcher, F. Kamiran, and M. Pechenizkiy, "Technologies for dealing with information overload: An engineer's point of view," *Information Overload: An International Challenge for Professional Engineers and Technical Communicators*, 2012, pp. 175–202.

[45] D. Hiemstra, "Information retrieval models," *Information Retrieval: Searching in the 21st Century*, 2009, pp. 2–19.

[46] T. Berners-Lee, J. Hendler, O. Lassila *et al.*, "The semantic web," *Scientific American*, Vol. 284, No. 5, 2001, pp. 28–37.

[47] M. Fernández, I. Cantador, V. López, D. Vallet, P. Castells, and E. Motta, "Semantically enhanced information retrieval: An ontology-based approach," *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 9, No. 4, 2011, pp. 434–452.

[48] H.J. Happel and S. Seedorf, "Applications of ontologies in software engineering," in *2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2006)*, 2006. [Online]. https://km.aifb.kit.edu/ws/swese2006/final/happel_full.pdf

[49] J. Scott Hawker, H. Ma, and R. Smith, "A web-based process and process models to find and deliver information to improve the quality of flight software," in *The 22nd Digital Avionics Systems Conference*, Vol. 1. IEEE, 2003, pp. 3–B.

[50] J. Caralt and J.W. Kim, "Ontology driven requirements query," in *40th Annual Hawaii International Conference on System Sciences*. IEEE, 2007, pp. 197c–197c.

[51] P. Inostroza and H. Astudillo, "Emergent architectural component characterization using semantic web technologies," in *Proc. Second International Workshop Semantic Web Enabled Software Eng.* Citeseer, 2006. [Online]. https://km.aifb.kit.edu/ws/swese2006/final/inostroza_full.pdf

[52] B. Antunes, P. Gomes, and N. Seco, "SRS: A software reuse system based on the semantic web," in *3rd International Workshop on Semantic Web Enabled Software Engineering (SWESE)*, 2007.

[53] A. Abran, P. Bourque, R. Dupuis, and J.W. Moore, *Guide to the software engineering body of knowledge – SWEBOK*. IEEE Press, 2001.

[54] C. Calero, F. Ruiz, and M. Piattini, *Ontologies for software engineering and software technology*. Springer Science & Business Media, 2006.

[55] P. Wongthongtham, E. Chang, T. Dillon, and I. Sommerville, "Development of a software engineering ontology for multisite software development," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 21, No. 8, 2009, pp. 1205–1217.

[56] O. Mendes, A. Abran, and H.K.M. Québec, "Software engineering ontology: A development methodology," *Metrics News*, Vol. 9, 2004.

[57] J.R. Hilera and L. Fernández-Sanz, "Developing domain-ontologies to improve sofware engineering knowledge," in *Fifth International Conference on Software Engineering Advances (ICSEA)*. IEEE, 2010, pp. 380–383.

[58] J. Radatz, A. Geraci, and F. Katki, *IEEE standard glossary of software engineering terminology*, The Institute of Electrical and Electronics Engineers, Inc. Std. 610.12-1990(R2002), 1990.

[59] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "GATE: An architecture for development of robust HLT applications," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 168–175.

[60] K. Bontcheva, V. Tablan, D. Maynard, and H. Cunningham, "Evolving GATE to meet new challenges in language engineering," *Natural Language Engineering*, Vol. 10, No. 3-4, 2004, pp. 349–373.

[61] D. Ferrucci and A. Lally, "UIMA: An architectural approach to unstructured information processing in the corporate research environment," *Natural Language Engineering*, Vol. 10, No. 3-4, 2004, pp. 327–348.

[62] R.K. Yin, *Case study research: Design and methods*. SAGE Publishing, 2013.

[63] W.G. Cochran, *Sampling techniques*. John Wiley & Sons, 2007.

[64] C. Robson, *Real world research*, 2nd ed. Oxford: Blackwell Publishing, 2002.

[65] I. Rus and M. Lindvall, "Guest editors' introduction: Knowledge management in software engi-

neering," *IEEE Software*, Vol. 19, No. 3, 2002, pp. 26–38.

[66] N.F. Noy and D.L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*, Stanford University, (2002). [Online]. https://protege.stanford.edu/publications/ontology_development/ontology101.pdf

[67] U. Dinger, R. Oberhauser, and C. Reichel, "SWS-ASE: Leveraging web service-based software engineering," in *International Conference on Software Engineering Advances*. IEEE, 2006, pp. 26–26.

Appendix A. Interview guide

A1. Introduction

– Present yourself
– Ask about recording and confidentiality

The subject of the research is Semantic-Web based Enterprise Knowledge Management system. The focus is on improving information retrieval capabilities in knowledge management systems. That is, we want to explore the benefits of semantic search in enterprise environments. What we mean by semantic search is using meaningful, complex queries instead of traditional keyword based search platforms (e.g. Google) and retrieving aggregated knowledge from different sources. The result set in the semantic search is actually extracted knowledge instead of a set of documents that contain the search string. The reason why we would like to conduct interviews is to understand how Ericsson employees gather implicit and explicit knowledge during their daily work and specify the role of internal collaboration tools in this process. That is, we want know if these tools can satisfy the needs of people to find out the existing knowledge.

The focus is on how you cope with problems related to information overload and finding information. The data that we will collect in this interview will be very important for understanding the problems about the current situation and the usefulness of the proposed system to solve the existing problems. We believe it will be a benefit for the organization if we can reduce the time spent on finding relevant information and hence reduce the redundancy of sharing information.

A2. General questions about background and communication

1. Could you please tell me about your roles and responsibilities? (also current projects, previous experiences, etc.)
2. Can you tell me how you share information or documents in your projects with team members and with other related departments, units, etc.?
   – How would you classify the types of information you share?
   – What kind of tool do you use for each type of information?
3. What kind of problems do you face about sharing or finding each type of information? In which of these information types do you think there is information overload and people spend too much time to access information?
4. How often do you use collaboration tools/information/documentation of Ericsson (give examples)? (scale: daily, weekly) What purposes do you use them for? What kind of information do you look for or do you share? (possible scenarios). Do you easily accomplish your goals in these scenarios?
5. Can you give me example search scenarios from your daily work? Do you find documents by browsing around? In which cases? Search string examples?
6. How would you like to filter?
   – SWEBOK knowledge areas and practices,
   – Software lifecycle phases,
   – Document types,
   – Organizational structure (based on projects, products),
   – Domain.
7. How would you evaluate your satisfaction with the search facilities in these tools? WHY?
8. What do you suggest should be changed or improved when it comes to searching?
9. What do you do if you cannot find the information you are looking for in these tools?
10. How often do you need go and talk to a person with expertise or experience, in order to gather knowledge (even if it is simply an abbreviation that you don't know the meaning of). In what kind of situations does this happen? What kind of information?
11. How do you find the person to ask about a given issue?
12. When you need to ask a question, do you first perform a search if someone already shared this information? If so, do you usually find it or not?

A3. Demo and evaluation

Present the semantic tool with its functionalities and show search scenario examples based on the loaded discussion forum pages within the system. Illustrate different search types (such as faceted search, browsing the ontology, filtering).

1. What do you think about the presented tool? How would you rate its usefulness? Why?
2. How is the experience different from what you are currently using? Why?
3. Do you think the speed of finding information can change with this technology? If so how much would it change if you had to rate them on a scale?
4. For which type of scenarios and information types?
5. What improvements do you think can be made?
6. Would you use it to find the related people to ask your questions (to gain implicit knowledge)?
7. Would you prefer to add tags manually for every information you share for more accurate results, or you would prefer it automatic like this?
8. What about a software engineering ontology, would you search based on software engineering processes, artefacts?
9. If you have to rate on a scale, what would you say about using a semantic system like this in comparison with the existing systems you have? Would you prefer this version? Why?
10. Do you think we have missed anything important that we can mention? Do you have anything else to add?

e-Informatica Software Engineering Journal, Volume 12, Issue 1, 2018, pages: 265–302, DOI 10.5277/e-Inf180111

A Literature Review on the Effectiveness and Efficiency of Business Modeling

Magnus Wilson*, Krzysztof Wnuk*, Johan Silvander*, Tony Gorschek*
*Faculty of Software Engineering, Blekinge Tekniska Högskola, Karlskrona, Sweden
magnus.wilson@bth.se, krw@bth.se, johan.silvander@bth.se, tgo@bth.se

Abstract

Background: Achieving and maintaining a strategic competitive advantage through business and technology innovation via continually improving effectiveness and efficiency of the operations are the critical survival factors for software-intensive product development companies. These companies invest in business modeling and tool support for integrating business models into their product development, but remain uncertain, if such investments generate desired results.
Aim: This study explores the effects of business modeling on effectiveness and efficiency for companies developing software-intensive products.
Method: We conducted a systematic literature review using the snowballing methodology, followed by thematic and narrative analysis. 57 papers were selected for analysis and synthesis, after screening 16 320 papers from multiple research fields.
Results: We analyzed the literature based on purpose, benefit, challenge, effectiveness, and efficiency with software and software-intensive products as the unit of analysis. The alignment between strategy and execution is the primary challenge, and we found no evidence that business modeling increases effectiveness and efficiency for a company. Any outcome variations may simply be a result of fluctuating contextual or environmental factors rather than the application of a specific business modeling method. Therefore, we argue that governance is the fundamental challenge needed for business modeling, as it must efficiently support simultaneous experimentation with products and business models while turning experiences into knowledge.
Conclusion: We propose a conceptual governance model for exploring the effectiveness and efficiency of business modeling to occupy the missing link between business strategy, processes and software tools. We also recommend managers to introduce a systematic approach for experimentation and organizational learning, collaboration, and value co-creation.

Keywords: business modeling, business model operationalization, effectiveness, efficiency, context-dependent, governance, software-intensive product development, literature review

1. Introduction

Software-intensive product development (SIPD) companies experience digitalization of their business environments. The embedded flexibility that software offers merges with the high-pace technology innovation, resulting in new business opportunities for creating and capturing value in digital business ecosystems [1, 2]. This has implications for the business model.

A business model is a blueprint for a company's business logic and a description how to manage and innovate the business. Central to a business model is how an organization creates, delivers, and captures value [3]. Business models can be seen as a set of choices and consequences of these choices (strategies and tactics) that impact the realizing organizations, business processes, products, and systems [4]. Business modeling in a business ecosystem is an activity based on transactions of activities geared toward value creation for all stakeholders [5]. Business modeling (BM) is also a practice that aims to analyze the business environment and acquire

insights to formulate and drive change, by adapting and aligning the business strategy with the execution to ensure value delivery for all stakeholders [6, 7].

Optimizing value creation requires profound understanding how the implemented business model (organization, business processes, and systems) interacts with products and stakeholders for value creation and value capture [8]. SIPD companies have a unique position for optimally (efficiently) creating the correct (effective) value for all stakeholders. Given that software is the main component in: 1) the tools for implementing and supporting core business processes; 2) developing the software product itself, and 3) integrating the product into the business ecosystem, SIPD companies could seamlessly adapt and integrate their products to their business model using business modeling [9].

The business model mediates the link between technology and a company's performance, but the literature is missing the studies which focus on the interdependencies between business model choice, technology innovation, and success [10], as well as differentiating the value creation and value capture analysis over individual, organization, and society level [8]. Several prominent authors emphasized the lack of coherence and clear focus in the business model literature [7, 11, 12]. In particular, there is a gap in understanding how BM interacts with software-intensive products in the digitalization transformation, and what effects BM have on increasing the effectiveness and efficiency of the SIPD companies and maximizing the technology innovation realization effects.

This literature study aims to address this gap by investigating what factors determine the effectiveness of BM, and if BM can act as an enabler for improvements in effectiveness and efficiency of SIPD companies. This study provides a software engineering perspective on how software and software-products enable value creation as the unit of analysis for BM. This perspective enables us to narrow the scope of the vast business model literature, as well as limiting the size of the study by defining a more precise context for analyzing effectiveness and efficiency, as affected by the on-going digital business transformation. Based on the literature review results, we present a summary of benefits and challenges associated with BM including reported impacts on the effectiveness and efficiency of the business. Next, we synthesize the implications for the research and practice of BM and propose a conceptual governance model (CGM) for exploring the effectiveness and efficiency of BM (addressing both the innovation of business models as well as the outcome on company level for the implemented business model).

The paper is structured as follows. In Section 2, we introduce fundamental concepts related to BM and theories used to investigate the multifaceted, cross-disciplinary view of BM and business models. Section 3 reports on related work to BM and its usefulness while Section 4 contains a detailed description of the study design and study execution including a validity discussion. Results are presented in Section 5, starting with general results around the study itself, followed by the detailed results regarding each research question. In Section 6, our research synthesis including trends and our proposed CGM for exploring BM are presented. Finally, in Section 7, we list six implications for researchers and industry followed by our conclusions and key statements in Section 8.

2. Background

2.1. Effectiveness, efficiency, and governance in BM context

Business modeling shares several similarities with software engineering, requirement engineering [13–15], and software product lines (SPL) [16]. Software engineering provides new possibilities to efficiently and effectively implement strategies agreed upon during business modeling activities [2].

The business model literature describes several concepts associated with effectiveness and efficiency. They are often adapted to specific contexts, e.g., organizational efficiency, manufacturing efficiency, operational efficiency, product development efficiency, and expressed as a value,

time or in financial terms as for costs, revenues, profits, and margins. By starting with an "umbrella definition" offered by Webster-Merriam on-line, we will discuss definitions suitable for SIPD companies and our study.

Effectiveness is *the power to produce the desired result.* Efficiency is defined as *the ability to do something or produce something without wasting materials, time, or energy: the quality or degree of being efficient (technical)*, but also as *the power to produce the desired result* causing some ambiguity between the two terms. Buder et al. differentiate between quality (effectiveness) and required effort (efficiency) [17]. Organizational effectiveness is discussed by Zheng et al. in combination with strategy and knowledge management, where they use the definition *the degree to which an organization realizes its goals* [18].

Effectiveness is often measured as the quality of the desired result and Frökjeaer et al., in their attempt to correlate usability to efficiency and effectiveness, they define efficiency as *[. . .] is the relation between (1) the accuracy and completeness with which users achieve certain goals and (2) the resources expended in achieving them* [19]. Measurements of efficiency are often related (direct and indirect) to time and cost. In economics the term efficiency focus on different aspects of the balance between supply and demand. It is measured by the relationship between the value of ends and the value of means and examples of terms are allocative efficiency (production represents customer preferences) and productive efficiency (cannot produce more of one good without sacrificing production of another).

Effectiveness and efficiency are subjective and depend on evaluations. Such evaluations are based on an individual's understanding of knowledge and interpretation in a specific context [20]. Therefore, having the same understanding of a context (which the measurements are relative to), is fundamental when defining effectiveness and efficiency measurements for BM (and the over-arching business context). Current research on context description in software engineering provides a useful checklist on context facets (product, processes, people, practices and techniques, and organization and market) [21]. Understanding, specifying, and sharing contextual factors (often as part of contractual agreements) is a critical factor for systematically optimizing the level of sub-optimization in a business ecosystem.

Effectiveness and efficiency are also closely related to governance, and Webster-Merriam on-line defines governance as *the way that a city, company, etc., is controlled by the people who run it.* Understanding governance is also a crucial part of BM as indicated by for example [5,22,23]. Jansen considers measurements and governance as the enablers of a successful software ecosystem [24]. Zott and Amit argue governance is a vital part of evaluating BM experimentation [5]. Page and Spira discuss corporate governance connected to the business model as a growing need to attain accountability by the board by considering conformance, performance, and overseeing management control systems. They conclude that corporate governance is essentially the same thing as sustaining and developing business models [25]. In this paper, we will use the Webster-Merriam definition of governance.

2.2. Business modeling as an enabler for a company's efficiency and effectiveness

There are many diverse and even divergent definitions of a business model and BM, as also highlighted in many literature reviews, e.g., [7,11,12,26]. A business model "models the business", but as such it has a wide range of usage depending on who and why is using it. It can be used as a description of "kinds and types" in a taxonomy to compare businesses or like a recipe for designing and innovating successful (new) business. Business models can also act as a description of the "logic of the firm", i.e., how to create value and generate profit, or as a scale model to investigate, analyze, and evaluate different strategies and tactics, thereby supporting both strategic and daily decision making [27].

There are two ways to interpret "efficient and effective." One interpretation is that the BM process itself should be efficient and effective. The other interpretation is that the business

model realization should increase a company's efficiency and effectiveness, i.e., BM should be the practice that increases a company's efficiency and effectiveness. In this work, we follow the second interpretation of efficient and effective, as we are primarily interested in BM as a way to enable improvements in a company's efficiency and effectiveness. Therefore, we base our work on the BM definition by Rohrbeck et al. as *to be a creative and inventive activity that involves experimenting with content, structure, and governance of transactions that are designed to create and capture value* [28]. This definition supports our investigation of BM for SIPD companies in two ways. Firstly, looking at value creation transactions allows for a value-driven business model analysis in a business ecosystem. Secondly, by introducing the word *experimenting*, it extends BM to a process of "translating an idea into execution, testing and changing until satisfied," similar to the agile software development methods. We complement the BM definition with the proposed capabilities needed for BM (understand and share, analyze, manage, and prospect) [9].

2.3. Translating business strategy into execution using business models

Casadesus-Masanell & Ricart argue a clear distinction between strategy and the business model, where the business model *is a reflection of the firm's realized strategy* and that the strategy is the plan and process to reach the desired goal, via the business model and onto tactics [4]. Among the authors that recognize the role of the business model in translating business strategy into execution, Doganova talks about the business model as a "calculative and narrative device" to innovate and translate the business strategy into execution [29]. In the same vein, Osterwalder defines the business model as a formal model to capture and translate a value-based business idea into requirements for the ICT systems and the organizations that execute that business model [9]. Höflinger defines *A business model is the design of organizational structures for converting technological potentials into economically valuable outputs by exploiting business opportunities* [7]. For this paper, we combine our transaction-based (bottom-up) definition of BM with Höflinger's (top-down) framework for defining the business model since:

- He extensively integrates and builds on the literature for business models.
- He addresses the issue of static versus dynamic business models (where he supports the static nature of the business model and argues business model innovation as the approach to adapt to rapidly changing environments).
- He focuses on the consequences regarding multi-value, superior performance and organizational learning as a mechanism for feedback and control.
- By taking an inside-out view of the research gap addressed in this study, i.e., based on how software and software-products enable value creation as the unit of analysis for BM, it enables both a top-down and bottom-up analysis.

Translating business strategy into execution is not an easy task and requires experimentation with content, structure, and governance of transactions that are designed to create and capture value [28]. Rohrbeck et al. advocate collaborative BM as a way to deal with the complexity and uncertainty of systems and markets. They stress the need for planning, decision making, validation, and experimentation in highly complex environments. Other scholars also acknowledged the role of experimentation in BM [30–32]. Experimentation can help to capture and manage the business environment dynamics, but it also implies new challenges in addition to just capturing and designing a business model. Some of these challenges are emphasized by Ballon when he argues *it is precisely the alignment of control and value parameters that is of most relevance to business modeling* in his aim to describe a theoretical foundation for operationalization (preparing for execution) of the business model [33]. Ballon proposes an analytical framework for making the scope for choice explicit while connecting value to the configuration of a business model, while others formulate the main challenge as *organizations have to reach the alignment state and maintain it alongside its evolution* [34].

2.4. Capturing the change dynamics and value with software products

Effectively dealing with change requires understanding how the concept of strategy relates to the business model and tactics [4], what strategic agility [35] and strategic flexibility [36] the organizations have, as well as how changeability (adaptability, agility, robustness, and flexibility) can be operationalized using modularity in design and software-based systems [37]. Flexibility and adaptability has since long been a top priority for CEOs[1] and business model innovation is becoming a top priority amongst CEOs[2]. Hence, an important part of analyzing efficient and effective BM translates to capturing and managing the change dynamics of today's business operations.

Value creation and value capture are the central concepts for BM. However, there is still missing consensus on the boundaries of these concepts, based on: (1) plurality in source and target; (2) mix of content and the process; and (3) the overlap between value creation and capture. Value creation is divided into use value (as perceived by an individual) and exchange value (as the monetary compensation), and should be related to the source and the target (individual, organization, and society). Value creation is highly subjective and context-specific but always rooted in interactions. Value creation should be primarily analyzed on the individual level, while most business model literature discuss value creation on the organizational level. Value capture overlaps value creation by discussing the sharing of value (value slippage) to society, organizations, and individuals [8].

Moore discusses value creation in a business ecosystem and the importance to have *value-in-the-experience of customers, economics of scale, and continuing innovation*, while investing in expanding communities of allies. He defines a business ecosystem as a complex structure of interested parties and communities interacting with each other to produce and to consume goods and services, in a partially intentional, highly self-organizing, and even somewhat accidental manner [38]. In such a volatile and increasingly complex environment, successful companies cannot just add value, but instead need to address the value-creating system itself. They must reinvent value, and work together with all stakeholders in the business ecosystem to co-produce value [39].

The flexible nature of software-intensive products opens up unique opportunities to quickly reinvent and co-produce value, but also presents new challenges for SIPD companies in business ecosystems [37, 40]. Figure 1 illustrates an example of software-based value creation in an ecosystem, highlighting three distinct, but overlapping process areas: (1) core business processes, (2) product development, and (3) product integration.

SIPD companies possess unique opportunities to harvest the flexible nature of software and reinvent value by integrating and developing native product support for each respective area and the business model(s). These areas are extensively discussed in the business model literature, e.g., covering pure software business models [41], open source/mixed source [42] and digital options [43], transitions from product-based business models to service-based models [44], or to industrial product-service systems and use models [37, 45, 46]. Even mechanical products rapidly become software-intensive products [47].

The software value map (SVM) [48] explores the different value perspectives and the challenges of balancing the relevant value aspects in software development. The SVM is an extensive collection of software value aspects categorized in four perspectives[3]: customer value; the financial perspective; internal business perspective; and the Innovation, market and intellectual perspective on value. The SVM puts precise and explicit terms on concepts discussed by Höflinger,

[1]Based on CEO Challenge 2004: Perspectives and Analysis, https://www.conference-board.org/publications/publicationdetail.cfm?publicationid=893, and revisted by http://www.floordaily.net/flooring-news/survey--most-ceos-say-flexibility-and-adapting-to.

[2]IBM's global CEO report 2006: Business model innovation matters, http://www.emeraldinsight.com/doi/full/10.1108/10878570610701531.

[3]See http://www.softwarevaluemap.org for the SVM Tool and latest details, as it is continuously updated by input from more than 50 companies world-wide, October 2016.

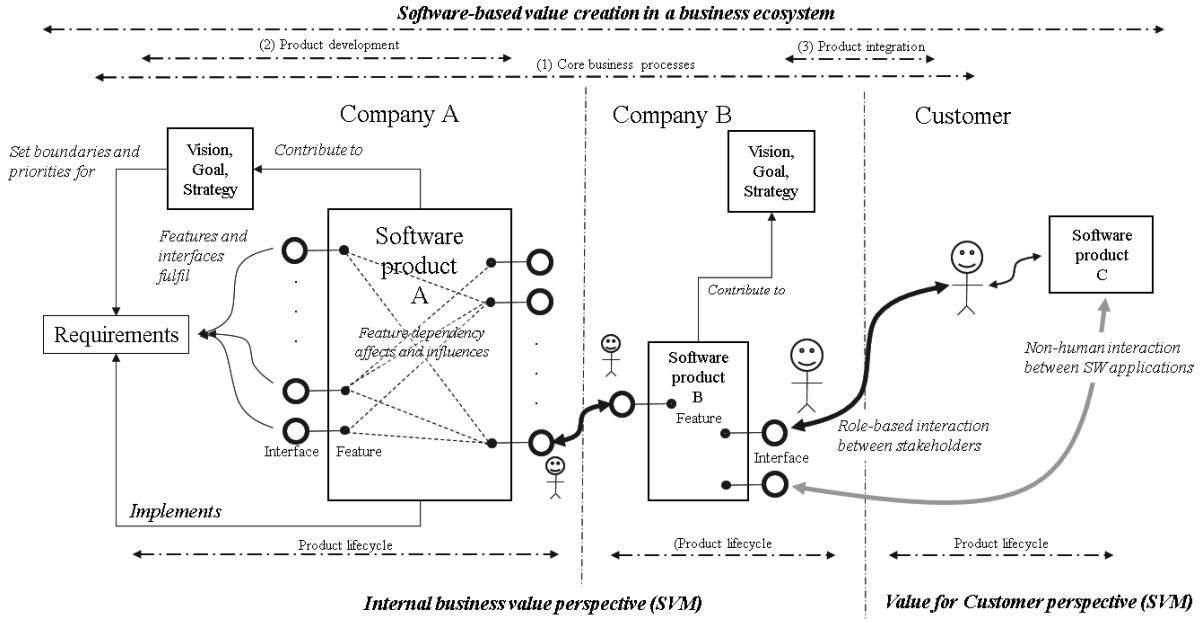


Figure 1. Software-based value creation in a value delivery chain in a business ecosystem

e.g., know-how characteristics, value structure, financial value, social value, and organizational learning. The SVM provides a necessary but often neglected bridge between product strategy, value, and operationalization of software systems and products in requirements elicitation, and decision making.

In Figure 1, two companies, and a customer interact in a business ecosystem. The software products are involved as agents via interfaces and features along the value delivery chain. Value is created in the interaction between two stakeholders, indicated by the arrows between the stick-men and their smiley faces. A company needs to look beyond their borders to identify all stakeholders and possible interactions for value creation (at society, organizational, and individual level).

Different aspects of value are created in these interactions, while external conditions and influences shape the perception of value (as technology and society advances), often resulting in a misalignment between expected and perceived value. BM (in a SIPD context) aims to systematically capture, prioritize, and address how business logic, resources, and governance should be operationalized for optimal value creation and value capture. A software product is hence an essential part of the operationalized business model, both by acting as an agent to the business model (the content, structure, and governance of transactions), as well as through optimizing a software product's changeability [37] to adjust for external influences.

Figure 1 also illustrates the recursiveness and complexity of business models and software-based value creation. Each company typically run their business model while the "overarching" business model for the business ecosystem can be seen as an aggregation and collaboration of the "underlying" business models [28]. Software Product C (e.g., a browser) is using Software Product B (e.g., a crowd-funding application delivered as a cloud service), which in turn is relying on Software Product A (e.g., a database application delivered as a service). Each company develops their software product(s) based on their (business model's) vision and goals. They constantly need adjusting for external influences, using requirement engineering to constrain the vision and goals into an "optimal" realization (time, opportunities, risks, features, and resources) of the software product. A software product should have features addressing (all) the needs of (all) stakeholders (throughout the complete value delivery chain). It must also

support any stakeholders' interaction with the software product throughout the product's entire life-cycle (from the idea, design, production, commissioning, usage, to de-commissioning and obsolescence). Such role-based interaction is illustrated in the figure with features, interfaces, bi-directional arrows and the stick-men. An interaction can also be a non-human interaction between two software products, entirely internal to a company, or any combination thereof. These interactions occur at all levels in activities between actors, within and across company borders, as well as within different life-cycles of the value delivery chain. In a business model, a transaction is an aggregation of such role-based interactions where the exchange of information, goods, payments, and feedback are not necessarily synchronized. Also, the different software products' life-cycles interact and overlap. This puts new requirements on the software product to more efficiently handle the introduction of new interactions and collaborations, e.g., customers being part of the design or test of Company B's software product while Company A and B enter a partnership agreement to share costs and revenue [28]. For SIPD, this creates a tight, highly recursive relationship between BM and the software products.

3. Related work

Several prominent literature reviews are published on the topics of business models. For brevity, we focus on recent publications highlighting aspects relevant for performance [7,11,12,49]. Common to all reviews is the lack of empirical evidence that using BM to evolve the business model increases a company's effectiveness and efficiency. Lambert and Davidson summarize 40 publications and report that choosing the right business model is one factor for a company's success based on evidence of a relationship between success, business models, and business model innovation. They conclude that the studies measure and report what is the current situation, but no empirical research aims to predict company success.

Three of the reviews [7,11,12] highlight the two major challenges in current research on business models: 1) that business model research is too dispersed and needs a consolidation of concepts; and 2) that it is difficult to connect strategy (via business model) to execution, while capturing and handling the needed dynamics of today's global and multi-stakeholder business environments. Other prominent researchers also highlight the lack of a consolidated body of knowledge and concepts [9,23,50,51], indicating a gap in understanding BM's real-world effects.

Business models for explaining a company's performance are frequently discussed both conceptually [52,53] as well as empirically [54–56]. Hacklin and Wallnöfer conclude that the business model acts more as a symbolic artifact and not as an analytic tool. Zott and Amit report empirical evidence suggesting that business model design can provide a competitive advantage, but does not provide conclusions that employing BM to evolve the business model will improve a company's effectiveness and efficiency. Lambert and Davidson studied the relationship between company success, business models and business model innovation. These studies all measure and report what is the current situation, but there is no empirical research that aims to predict company success or to conclude that business modeling enables effectiveness and efficiency of a company [49].

Osterwalder et al. advocate formalization of business models using IS/IT tools and an experimental approach "when-and-how-to-build" [57]. Their eight propositions to be observed and eventually tested seems still be equally valid: 1) use rigorous meta-models; 2) increase understanding business and IS/IT; 3) improve integration business and IS/IT; 4) facilitate and improve IS/IT choices infrastructure/applications; 5) facilitate choices IS role and structure; 6) help defining company's goals; 7) facilitate identification of key indicators; 8) externalize, map and store knowledge of value creation logic [9].

Giessmann et al. extend Osterwalder et al.'s propositions to build a model that can analyze and compare business models, but their work does not address the issues of aligning and daily

execution of a business model [58]. Salgado et al. also build on Osterwalder's business model canvas (BMC) and discuss how to generate a BMC from business goals, rules, and processes, but do not further connect the results to the IS/IT realization and daily operations [59]. They also discuss the alignment between business and IS/IT (from the lens of business model artifacts, enterprise modeling, and strategy and goal modeling) and formulate the main challenge as *Achieving alignment per se is not enough, organizations have to reach the alignment state and maintain it alongside its evolution* [34].

The literature indicates a research gap between modeling the business and executing the business model and more specifically, do business modeling increase a company's effectiveness and efficiency? Höflinger's framework extensively builds on the literature but does not empirically define or explore his angle of *superior performance*, nor the dynamics of a business model related to value. Further, he does not explore how the learning of an organization interacts with the design of, the representation of, and experimentation with a business model [7]. Rohrbeck et al. stop at the preparation for development and do not provide further insights into the mechanics needed for actual experimentation and validation of a business model [28]. Richter et al. discuss flexibility and value as a way to deal with change and implementation of business models. They conclude that further work is needed to better understand inter-firm governance structure [37]. Ballon proposes an analytical framework for making the scope for choice explicit and concludes that further work is needed to make interdependencies of parameters explicit and to extend the model in a more prospective and predictive sense [33].

4. Methodology

4.1. Research questions

We used software and software-intensive products as the unit of analysis. The rationale comes from the central role that software-intensive product play in the on-going business environment digitalization transformation. We focus on the following two research questions:

RQ1: *What benefits and challenges of business modeling are reported in the literature?*

RQ2: *What effects related to effectiveness and efficiency of business modeling are reported in the literature?*

We used RQ1 to investigate the contextual setting for business modeling and to compare and analyze the reported effects on efficiency and effectiveness. The on-going business environment digitalization transformation heavily depends on flexible and scalable software solutions. Therefore we limit the scope to business modeling for SIPD companies developing software-intense products and services. The research process executed in this study is outlined in Figure 2.

4.2. The snowball methodology

Our systematic literature review (SLR) methodology is based on the guidelines for snowballing literature search proposed by Wohlin [60]. The snowballing methodology is considered less noisy compared to a similar database-search based methodology and the critical step for a successful snowballing is to choose a good tentative start set characterized by: 1) studies from different communities; 2) size appropriate for the studied area; 3) diversity of publishers, years, and authors; and 4) is based on the research questions and keyword. The complete study was conducted in four steps, outlined in the subsections below and depicted in Figure 2. We screened 16 320 papers resulting in 57 papers included in the study.

4.2.1. Step 1: Design of the literature review

To minimize the author-bias and to prepare for a cross-disciplinary study (business management and software engineering), we performed two open-ended interviews to identify further reading to understand the terminology to formulate our research questions. These interviews helped us to decide upon the methodology, validity risks, inclusion criteria (IC) and data extraction properties. We also created a study protocol and documented each step and decision. The same IC

Table 1. Search strings for start set

Id	Terms
SS1	("business model" OR "business ecosystem") AND "value creation" AND "strategy"
SS2	("business modelling" OR "business modeling" OR "business ecosystem") AND "business strategy" AND "value creation" AND ("effectiveness" OR "efficiency" OR "business flexibility" OR "modularity" OR "variability in realization" OR "governance" OR "multi-business")

were used defining both the start set and in the following snowball iterations, see Appendix B.

4.2.2. Step 2: Defining the start set

We used a database search in Google Scholar to find the start set and recommendations from the interviewed experts. The two initial interviews (60-minutes, open-ended interview with the question *Does business modeling enable improvements in effectiveness and efficiency for a company?*) with experts in software engineering (telecommunication industry with 25 years of experience) and business management (professor in production management) resulted in a starting point of:

- four recommended studies, of which Höflinger also ended up in the start set [7];
- a wide multi-disciplinary map of subject areas: computer science; software engineering; business management and accounting; economics, econometrics and finance; organization management; and decision science;
- additional keywords – open innovation, strategic management, value creation, value capture, flexibility, business model innovation, business ecosystem, organizational theory, knowledge management, service science, enterprise architecture, software product lines, open source, and product service systems.

After further search in Google Scholar for definitions on these keywords, we created a recommended Golden Set (31 papers) from which we derived a collection of definitions to help us penetrate the terminology. The snowballing methodology recommends using Google Scholar to avoid any bias on specific publishers [60]. The definitions helped us develop the search strings (SS). We used a traditional search schema with iterative clustering to reduce the number of hits while minimizing noise (initially in Scopus since it contains all the subject areas). We ended up with two search strings[4], see Table 1, used to query six databases, see Figure 2.

Executing SS1 and SS2 (limited to title-abstract-keywords) resulted in 2948 papers, see Figure 2. The first author applied the inclusion criteria on titles and abstracts, and 2378 papers were removed. The remaining 570 papers were put in an excel sheet so duplicates and not peer-reviewed papers could be discarded. The final 477 papers were screened more thoroughly (abstract, introduction, conclusion) for IC and the result discussed and validated with the second author, leaving nine papers to be included in the start set. One paper recommended by the experts in business management was also included in the start set.

4.2.3. Step 3: Execute snowballing iterations

The first author collected the references of citations to the papers selected in each iteration. Next, we applied inclusion criteria and calculated the Cohen's Kappa in all iterations, see section 4.3.

In total, we screened 10 414 citations and 2958 references, see Figure 2. Iteration 1 covered the start set and resulted in 35 selected studies (out of 612 references and 249 citations). Iteration 2 resulted in 2011 references and 10 134 citations. The noise in citations is one of the downsides reported for the snowballing methodology, and we applied an initial pre-screening (language, title, abbreviated abstract) giving us a remaining 1335 citations to screen. By having the candidate list in Excel, it was easy to detect all duplicates. We selected 11 studies in iteration 2. Iteration 3 rendered 313 references and 30 citations resulting

[4]SS1 uses stemming and SS2 doesn't. Also, "multi-business" was added upon recommendation of industry expert, since executing several business models in parallel is a significant challenge for large SIPD companies.

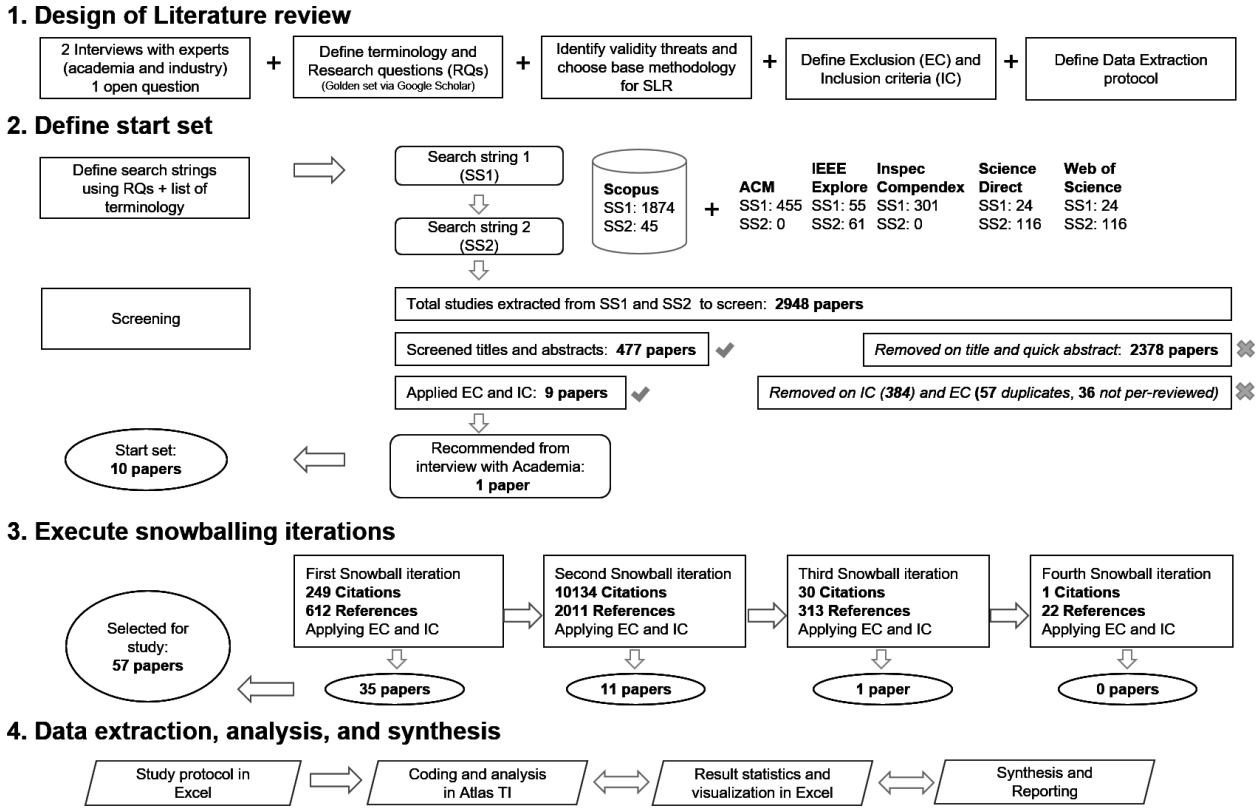


Figure 2. Research methodology overview

in one new paper selected. We got a natural stop of the snowballing procedure by iteration 4 with no more studies discovered resulting in a total of 10+35+11+1 = 57 studies selected for analysis, see Appendix A for a complete list.

4.2.4. Step 4: Data extraction, analysis, and synthesis

Appendix C outlines the data extraction properties (EP) used in this study. ATLAS Ti[5] and Excel were used to keep track of and analyze results as well as synthesize extracted information. The extraction was done by the first author and validated by the other authors, see section 4.3.

Properties EP1–EP4 were evaluated per paper and used to analyze the relevance to industry for each paper's contribution. The property EP3 (Rigor & Relevance) was also used for quality assessment, see extracted raw data per paper in Appendix A and detailed calculations in Appendix C. It helped us to evaluate generalizablity of the results, see section 4.3. Open coding [61] was used for properties EP5–EP9 and the extracted data was thematically analyzed. Properties EP5–EP9 helped us synthesize results regarding BM as phenomena as well as to identify potential research gaps.

The results were iterated in two phases (a) RQ1 and (b) RQ2. For each phase, the first author prepared a summary of listed quotations from all studies. The list was then reviewed against the extracted result, and the first author had to explain a summary of each paper's findings to the reviewer. Phase (a) were reviewed by the second and third author, while phase (b) were reviewed by the second author.

4.3. Validity threats

We adopted the validity guidelines suggested by Runeson et al. [62]. An extensive industrial experience of the authors may have influenced the aims of the study with a stronger bias towards

[5]Software for Qualitative Data Analysis, http://atlasti.com/.

solutions. We mitigated that bias by two initial interviews and an iterative refinement of the research questions and also by applying a grounded theory approach [61], fostering a focus on the merits of each paper before an end-to-end perspective could be evaluated.

The selected ten papers in the start set are highly heterogeneous and therefore minimize the bias on specific author or terminology. Similarly, we mitigated the author's bias by calculating the Kappa coefficient when selecting the start set papers. The Kappa analysis was done by the first and second authors, and the value was $\kappa = 0.566$ and later increased to $\kappa = 0.638$. The Kappa analysis was also performed during the first snowballing iteration on 12% of the studies with a resulting value of $\kappa = 0.763$. These values represent sufficient agreement and increase the validity of the study.

To mitigate author bias during extraction, six random studies were selected (of the 57 studies) and extracted by the first and second authors. The validation showed a discrepancy of one paper for extraction properties EP1–EP4 and after further discussion full agreement was reached. Also, the results to the RQs (EP5–EP9) was iterated in two phases, and each phase was presented by first author before discussed and evaluated by at least one more researcher.

Rigor and relevance analysis was applied to mitigate potential threats to conclusion validity. The rigor classification based on software engineering literature was also adapted for business modeling literature. The relevance parameter was coded using binary weights (0, 1, 2, and 4 instead of the recommended 0 and 1). We also decided to add property EP4 to specifically address the relevance of a paper's content concerning our RQs (since the property EP3 and its' relevance aspects only consider the research method and context of a paper). This provided higher resolution when discussing the relevance and when thematically comparing the papers. The extraction of results was iteratively reviewed and discussed with second and third authors. We minimized potential internal validity threats by following the systematic mapping study guidelines, creating a review protocol and sharing the work associated with data extraction and analysis.

Since this study covers studies from a wide set of research fields, the semantics (and context) of words can often be misleading. We addressed this by our choice of a snowballing methodology in combination with a rigor design to identify the start set. Moreover, we used open coding (inspired by grounded theory [61]) to synthesize and harmonize language between the different research fields.

Because of the interdisciplinary nature of this study, the risk remains that some aspects are underrepresented and other aspects are overrepresented. In particular business model innovation or business process modeling seems to be heavily researched in the business management and the computer science community. However, we decided to limit the scope in these dimensions since our primary interest is the interplay between the strategic intentions, the design of a business model, the realization of it, and the resulting effects on efficiency and effectiveness, rather than details on how individual steps are performed.

We selected our start set studies from different research disciplines and these studies are conducted using many different research methods which improve the external validity of our literature review. Even though the start set is carefully chosen and includes publication years (2004–2014) there are only 17 (out of 57) papers published during 2013–2015.

5. Results and analysis

Table 2 shows results related to research questions mapped to each paper's context (data extraction property EP4, see Appendix refapp:C), including frequency and summarizing comments. Using inclusion criteria IC2 and IC3 we investigated if the papers address flexibility without further exploring the efficiency or effectiveness.

74% of the identified studies (EP4, categories 2 and 3) focus on the business model construct rather than the BM as a practice. One reason for this could be that BM as a practice is a broad, diverse topic forcing researchers to limit the scope

Table 2. Results mapped to research questions and paper context

RQs /ICs	Business modeling (1)	Business model (2)	Other (3)	Sum of papers	Comment
RQ1	2, 6, 15, 17, 18, 35, 36, 37, 41, 49, 51, 52, 53, 54, 56	1, 3, 5, 7, 9, 13, 14, 16, 19, 20, 21, 22, 24, 29, 32, 33, 39, 40, 45	8, 10, 12, 26, 30, 31, 34, 38, 42, 43, 46, 48, 55, 57, 58, 59	50	Scattered in a multitude of practices and frameworks. Results suggest lack a systematic alignment of contextual information hindering re-use and integration of practices
RQ2	17, 35, 37, 54, 56	1, 5, 24, 29, 32, 45	8, 42	13	Quotes on effectiveness and efficiency are not differentiated nor substantiated
IC2	2, 6, 17, 18, 35, 36, 37, 41, 49, 52, 53, 54	1, 3, 5, 7, 9, 13, 14, 19, 20, 22, 24, 27, 29, 33, 39, 40, 45	8, 10, 12, 26, 30, 31, 34, 38, 48, 55, 57, 58, 59	42	Many papers reflect over flexibility. Governance is important for understanding the value (and cost) of (the right) flexibility in order to optimize the value creation and value capture
IC3	2, 6, 15, 18, 35, 37, 49, 51, 52, 54, 56	1, 3, 5, 7, 9, 13, 16, 19, 21, 22, 24, 27, 29, 32, 33, 45	10, 12, 26, 31, 34, 43, 46, 55	35	Variability in the realization is an important aspect of flexibility and should be a part of the business modeling analysis
Sum of papers	15 (29%)	20 (39%)	16 (31%)		The % is calculated of the 51 papers adressing RQs+ICs. 6 papers of the total 57 selected papers did not specifically address any of the RQs+ICs. They all belonged to category 3: Other
Hit rate	33% (5)	30% (6)	9% (2)		The 'hit rate' is the ratio of papers addressing both RQs. For category 3 the ratio include the 6 papers (not listed in the Table) not addressing any RQs

by addressing some aspects of a business model construct rather than BM as an activity or process. Still, only 33% of the paper address both RQ1 and RQ2.

The number of papers addressing multiple RQ+IC is growing since 2005. As the area becomes more mature, it is also becoming more complex, multifaceted, and cross-disciplinary. This trend is also indicated by Kindström where he states *that companies need to focus on all areas of their business models in a holistic fashion, and not just change isolated elements* [P24]. Similar, Reim et al. concludes that more research efforts are needed on the complicated relationship between strategic and operational levels [P3]. This could be one of the reasons why business model research is still scattered and disperse. To evaluate BM efficiency, it is therefore essential to connect the business strategy via the business model to the execution of the business model with traceability to daily operations and results.

We used Rigor and Relevance (EP3) to analyze the identified papers, see Figure 3 and Appendix A. 60% of the studies received industry relevance scores greater than 7, representing a good balance between state-of-art and state-of-practice. A majority of these studies (20) score 15 (highest), and additional eight studies score > 9 (two or more conditions met). The included literature reviews [P3, P9, P29, P40] have (as expected) a relevance score $= 0$ with acceptable rigor scores ($>= 1$). The remaining 19 studies with a non-industry relevance score, discuss specific topics or more general frameworks and methods/aspects (related to BM) divided on: strategy [P15, P19]; life cycles [P25, P28]; effectiveness and efficiency [P35]; flexibility [P27]; static/dynamic [P14, P34]; or frameworks, meth-

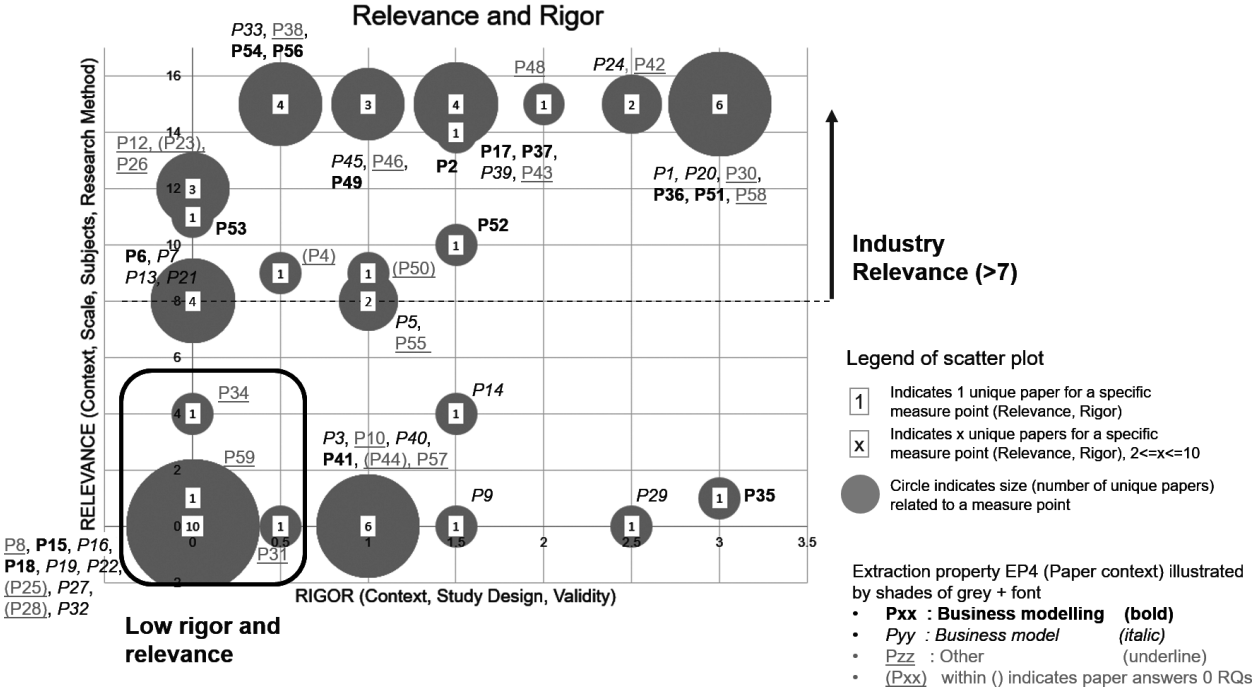


Figure 3. Papers plotted for frequency (size), rigor (X-axis) and relevance (Y-axis) scores, and paper context (font)

ods and models [P8, P10, P16, P18, P22, P31, P32, P41, P44, P57, P59].

45% of the studies are coded with a low rigor (score 0 and 0.5) where 11% only describe the context, but not mentioning any design or validity aspects. The validity aspect is the single most lacking aspect lowering the rigor in 54% of the 22 studies with medium rigor (score 1, 1.5 and 2). Different research fields are different regarding maturity, methodology, and best practices on how to report the research, which we believe are the main reasons affecting the rigor aspect.

5.1. Benefits and challenges associated with business modeling (RQ1)

We extracted 263 quotes of purpose, benefits and challenges of business modeling (EP5), see Appendix D. Quotes of purpose (P) often sets the general context, while quotes of challenges (C) or benefits (B) often are reflections of how well a solution to a specific problem works. Benefits refer to a solution with good enough result while challenges refer to potential issues to obtain a satisfactory result (judged by specific qualities and contextual factors). We identified the following common areas (rows in Appendix D): 1) value creation/capture; 2) cost/revenue; 3) mind-set and knowledge; 4) means[6] (mission, strategy, tactics, directives, organization, and resources); 5) ends[6] (vision, goals, and objectives); and 6) assessment[6] (decision control, clarity, visualization, influencer, etc.).

Our literature review results suggest that the overarching purpose found for BM is for a company to stay competitive and improve its business results. The quotes of purpose are often overlapping and cover a wide variety of more specific topics, like managing individual business aspects (e.g., offerings, market, cost and revenue), capturing the business logic and activity systems, over to a holistic nature like "operationalize strategy", appropriate value from technology, or managing value (co-creation, capture, creation) and partners. Investigating the quotes further, we identified three primary contexts for BM (columns in

[6]We use the terms assessment, ends, and means as defined in 2015 by Business Motivation Model Specification Version 1.3. http://www.omg.org/spec/BMM/. Accessed 2 Nov 2017.

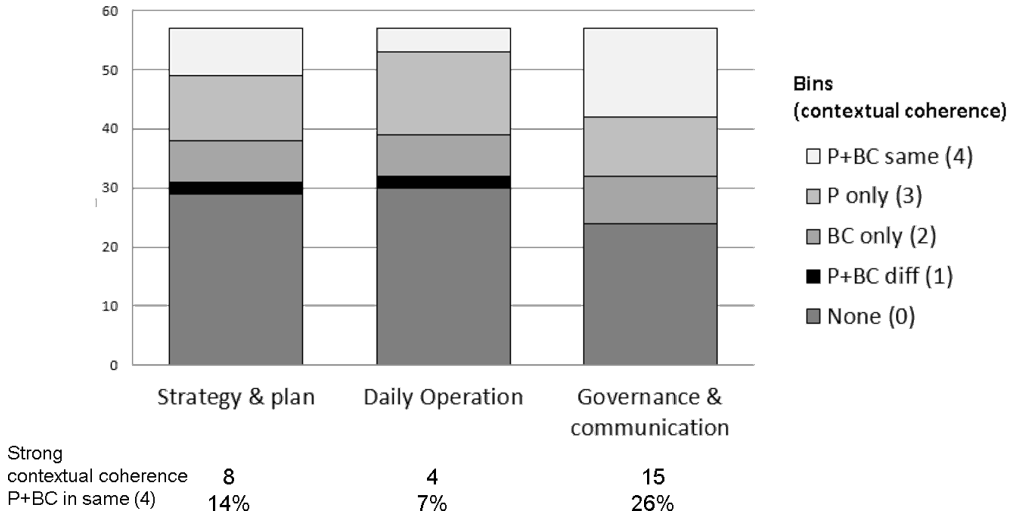


Figure 4. Quotes binned on purpose, benefits+challenges, and distributed over the primary contexts

Appendix D): 1) Strategy and planning; 2) Daily operations (executing strategies and plans); and 3) Governance and communication.

To analyze potential ambiguity (per paper) between the primary context of purpose quotes vs. the primary context of benefits/challenges quotes, each quote is tagged with Paper ID, Type of quote (one of P, B, C), and primary context (one of 1, 2, 3). Figure 4 illustrates the number of papers adhering to different contextual coherence bins distributed over the three primary contexts. We define the five contextual coherence bins. Bin 0 equals a paper having zero quotes in a primary context. Bin 1 equals a paper having quotes of P and B/C only in different primary context. Bin 2 equals a paper having only quotes (B/C) for a primary context. Bin 3 equals a paper having only quotes (P) for a primary context, and Bin 4 equals a paper having quotes of both P and B/C in same primary context.

Strongest contextual coherence is found in bin 4, with the highest ratio for the primary context "Governance & communication" at 16% (15 papers). The most significant contextual ambiguity (bin 1) is found in 4 papers [P8, P13, P19, P49] where a purpose is stated in one primary context while the benefit or challenge is claimed in another primary context without specific detailing the relationship. Romero & Molina discuss the purpose of value co-creation, as a complicated cooperative process (speed, coordination, compromise) with the challenge of managing the experience-sharing network, and how that affects the business modeling [P8]. Chesbrough discusses business model innovation with purposes related to formulating competitive advantage, value proposition and value chain definition while concluding challenges as a lack of tool support and continuous learning associated with BM experimentation [P13]. Richardsson discusses the purpose of formulating and achieving goals and objectives while concluding challenges as managing the different abstraction levels towards execution and getting the details right [P19]. Eurich et al. discuss the purpose of transforming the business opportunity into an organizational implementation via experimentation and business model fit, while concluding challenges in practical aspects like lack of details, not aligned design processes, disregard of external influences, etc. [P49]. Moreover, a significant portion of the papers lack statements on purpose, benefit, or challenge making a discussion around effectiveness and efficiency more challenging due to vague contextual information. Our results highlight a challenging issue how to effectively and efficiently defining contexts to improve understanding and communication in BM literature.

The importance of contextual information is mentioned by seven studies [P8, P17, P18, P20, P25, P51, P59], but no author goes as far as to suggest how to describe or represent the contex-

tual information. At the same time, the current research on context description in software engineering provides a useful checklist on context facets (product, processes, people, practices and techniques, and organization and market) [21]. However, these context facets are ambiguous in themselves, e.g., a market consist of products, customers, and organizations, a product could be a service and therefore include a process, etc. As a reflection of the identified challenges and claimed benefits, related to the paper's contribution to practices and methods for BM (including effects on effectiveness and efficiency), the underlying purpose is contextually vague with statements like "operationalize strategy" [P36, P37], or "deal with uncertainty" [P2, P52, P54]. The papers offer no empirical evidence to support that the purpose can be realized with claimed benefit nor do they quantify the extent of the challenges.

Similarities between the quotes on benefits and challenges are found, but only eight quotes are reported by multiple authors, for example: "(−) difficult managing dynamics (agility, adaptability, planning, decision) for alignment to environment and other organizations" [P2, P5, P7, P9, P36]; "(−) hard to visualize, document and share" [P26, P32]; "(−) difficult to mobilize and align available resource in time" [P9, P15]; "(+) better understanding, better language and legitimacy" [P17, P32]. We speculate that this low level of coherence between the papers is a result of the wide topical area of BM. We also note that seven of these eight quotes discuss common topics of governance ("handle dynamics", "align") and knowledge ("understanding", "sharing", "legitimacy", etc.), while the remaining statement covers value creation.

There are also cases where the same type of statement is argued both as benefit and challenge (by different authors). For example, (+) "building better strategies" [P32] vs. (−) "BM design requires better integration with strategy analysis" [P37] or (+) "improves dealing with uncertainty" [P2] vs. (−) "difficult to deal with uncertainty, complexity and dynamism" [P54] or (+) "improves alignment of strategy, organization and technology and integration business IS/IT domains" [P32] vs. (−) "hard to reach and maintain alignment of business model and information system model" [P59]. This kind of ambiguity can be a result of the wide topical area of BM in combination with a poorly specified contextual setting, opening up for a different interpretation of results.

The majority of the quotes are found in the union of (Governance) | (Mindset, Knowledge) | (Assessment) indicating that learning (knowledge) and control (governance) is key to BM. This is also backed by [P5, P13, P32, P51] which discuss the importance of experimentation and learning to adapt to the changing environment. The changing environment is also highlighted by [P2, P9, P49] as a challenging fact of business models, and as McGrath concludes, everything cannot be planned, but rather adapted to a suitable fit [P18]. In the same vein, we notice the vast number of papers belonging to bin 0, 2, and 3, indicating that a majority of the papers foucs on a single primary context of BM, rather than connecting the strategy to the execution and evaluating the business outcome (as a consequence of the BM practice).

Summarizing the results, the most common challenge is how to deal with the dynamics of business models [P2, P5, P7, P9, P36] and most of the quotes on challenges relate to the non-existing solutions for governance (representation, simulation, decision-support, and feedback) of the proposed frameworks and methods. Since governance is not addressed, each BM method or framework may work in its' specific context, but taken out of context or combined with other methods may fail to deliver the claimed benefits. Also, the quotes of benefits are unsubstantiated or claimed with limited empirical evidence (except for an empirical case which evaluates and compares user's understanding of two value models [P35]).

5.2. What impact does BM have on effectiveness and efficiency (RQ2)?

Only two studies make a clear distinction between the terms effectiveness and efficiency [P5, P35] where Chew and Buder & Felden both specifically link effectiveness to quality and ef-

ficiency to effort to perform a task. Zott et al. recognize efficiency as an important value driver, and that any value driver can enhance the effectiveness of the other drivers [P29]. Osterwalder et al. connect efficiency to infrastructure management while effectiveness is indirectly connected to value [P32]. Chew and Romero & Molina connect effectiveness to customer experience [P5, P8]. Mason & Mouzas argue efficiency is a product of careful management of resources and capabilities driven by a "network focused" approach while effectiveness (via marketing) is a product of being market-focused to keep in touch with changing customer needs by flexible products and service offerings [P58]. The terms are also used on different abstraction levels hindering in-depth analysis. We believe this is a likely result due to the combination of: 1) none of the 57 studies have research questions that directly address effectiveness and efficiency; 2) that business model research is still not coherent with a consolidated view of what a business model is used and useful for; and 3) few scholars address both primary contexts of strategy and the execution making an evaluation of effectiveness and efficiency difficult.

Measurements of effectiveness, efficiency, and company's performance (as an expected outcome of efficiency and effectiveness improvements) are neither sufficiently described nor substantiated. Measurements of effectiveness were only explicitly defined by Buder & Felden where they used a ratio of correctly answered questions to evaluate the effectiveness of individual methods about understanding value [P35]. No explicit measurements on efficiency or company's performance were found amongst the papers, except for Andries & Debackere who suggested company's survival rate to measure its performance for new technology-based business models [P42]. Ghezzi discussed how discontinuity can be detected before it affects a company's performance but does not mention how to measure the performance [P37]. A company's performance is also referred to by different terms but not further substantiated, for example by profitability [P29], value creation [P29], organizational performance [P29], operating cost or gains in productivity [P54]. We found no empirical evidence (except [P42]) to substantiate claims on effectiveness and efficiency. We also note that all 13 papers addressing RQ2 also address aspects of flexibility and variability in the realization (IC2 and IC3, see Table 2).

Indirect effects on effectiveness (and efficiency via profitability) are reported by three papers [P24, P29, P37]. Kindström discusses the transition to the service-based business model as a key to remaining competitive [P24]. He does not make any specific claims about effectiveness or efficiency, but proposes focusing research efforts on: 1) how to industrialize service offerings to a larger scale; and 2) understanding how a transition to service-based business models affects profitability and growth. Zott et al. in their literature review acknowledge the possible contingent effect of BM linking product market strategy and company performance [P29]. They also refer to a study by at IBM Global Business Services in 2006 that says financial out-performers put twice the effort on business model innovation compared to under-performers, but do not further elaborate as on how. Ghezzi looks at the strategic planning process and BM under discontinuity [P37]. He concludes that the 'business model parameters mix', as derived from the different business model blocks, directly affects the company's performance. He provides a strategy-analysis tool based on BM, VN, and RM constructs (business model, value network, resource management), to detect what is changing in the company's strategy when discontinuation occurs, but he does not discuss in any detail how to derive any changes in effectiveness or efficiency.

Summarizing the results, we found limited empirical results indicating that BM has an overall effect on a company's results regarding effectiveness and efficiency improvements. It is also not possible to judge whether a favorable outcome can be achieved in a scenario of continuous (experimental) BM, or it is just a result of a one-time activity to modify the business model. Also, we note that all 13 papers addressing RQ2 also address aspects of flexibility and variability in the realization. These limited results prompt us to do a contextual analysis of the effectiveness and efficiency of BM.

5.3. Contextual analysis of effectiveness and efficiency

We base our analysis on the two main contextual BM settings: 1) the business model realization should increase a company's effectiveness and efficiency; and 2) the effectiveness and efficiency of the BM process itself.

For **increasing effectiveness and efficiency** (contextual BM setting 1), we found the same three primary contexts as reported in Section 5.1: 1) strategy and planning; 2) daily operations (executing strategies and plans); and 3) governance and communication, see Table 3. From these contexts, we identified three patterns (full, partial, and single) describing whether a paper covers all three contexts or parts of them. The patterns are derived from the first three columns (define, execute, and governance) in Table 3. Full means that the paper does address topics in planning and strategy, daily execution, plus governance and communication contexts. Partial refers to any combination of two contexts, while single refers to only one context. We also analyzed the papers according to the three key areas aggregated from the studies: value creation/capture; decision support; mindset and knowledge.

The **BM process' effectiveness and efficiency** (contextual BM setting 2) are discussed by 3 of the 13 studies [P35, P54, P56]. Buder & Felden recognize the hurdle of keeping models consistent during transformations and suggest a specific value representation model as a remedy [P35]. Salgado et al. propose a method for modeling and visualizing requirements on the define and execute processes of the business model [P56]. Both studies offer limited empirical evaluations. Meier & Bosslau recognize the importance of a continuous, integrated BM to capture the dynamics of the ecosystem [P54]. It is the only paper clearly discussing the importance of not separating the process of BM from the actual define and execute processes of the business model. However, they do not quantify any effects on effectiveness and efficiency, while concluding that tools are a necessary focus for further research. We believe the lack of empirical results is a direct consequence of: 1) the wide contextual settings for business model research; and 2) the lack of consolidated view on what a business model is used and useful for. Given our study's primary focus (contextual setting 1), we also interpret the ratio of papers addressing our main contextual setting (77%) as a quality measure of our study design.

Full pattern category papers [P1, P5, P8, P24, P29, P54] advocate that to yield effectiveness and efficiency, the overall focus is how the plan/strategy/goal should be aligned with the execution of the strategy. Woodard et al. discuss how "design moves" enable rapid product development in a new domain with fierce competition and how to formulate and execute digital business strategies (align strategy to execution) based on option value and technical depth [P1]. They propose decision-support via option value and technical depth to integrate the perspectives of designers and corporate strategies. They empirically illustrate effectiveness and efficiency from a set of design moves but do not state on what level anything became more efficient.

A transition into service-based business models to improve competitiveness and efficiency of the business model is proposed by three papers [P5, P54, P24]. Chew argues that business model design impacts directly financial performance but does not state how nor to what extent it affects effectiveness [P5]. Effectiveness is a result of service variability and aligning the three contiguous processes for optimal value co-creation (customer value-creating, supplier value-creating, and the service encounter processes). He focuses on the **define** process with a service design concept to understand the customer needs and value appropriation, and concludes that execution also requires *support by a corresponding modular organizational architecture as well as IS architecture.* Meier & Bosslau discuss the difficulties when transitioning from a product-centric business model into a product-service centric model, with empirical findings that only 21% of manufacturing companies succeed in this transition [P54]. The fundamental problems are: a drop in efficiency, diversified portfolio, and an increased cost due to an increased product-service port-

Table 3. Identified effects on effectiveness and efficiency

Pattern and key areas	Strategy & planning (Define) (contextual setting 1)	Daily operations (Execute) (contextual setting 1)	Governance & communication (contextual setting 1)	Business modeling (contextual setting 2)
Full pattern P1, P5, P8, P24, P29, P54	x	x	x	P54
Partial pattern P32, P37, P56 P42	x x	– x	x –	P56 –
Single pattern P17, P35, P45	x	–	–	P35
Value creation/ Value capture	Concept of design moves [P1] Service concept design, service design, customer experience design, service architecture design [P5] Effective product market strategy [P29] Business process modeling efficiency [P35] Cumulative changes have a positive effect on success rate in immature markets [P42]	Concept of design capital [P1] Adaptations to initial BM are crucial, over- and under-adaptations effect performance [P42] The availability of resources and capabilities are more important to quality of adaptation [P42]	Transition to service-based business model improves profitability [P24] Dynamic business models (with flexibility) are important for a successful transition to service-based business models [P54]	Modeling overhead in transformation and reduction to maintain consistency [P35]
Decision support	Provide relevant information for next stage [P17] Strategic tools, business model, value network, resource management, signal radical change [P37] Empirical findings on instrumental efficiency for business modeling show no convergent results [P45] Process, goals, rules improves traceability [P56]	Decision-support via option value and technical depth [P1] Representation of information to enhance pragmatic validity [P17] Foundation for improved speed to react on external event and business environment [P32]	Quantitative modeling and simulation is vital in continuous loops [P54] Process, goals, rules improves traceability [P56]	–
Mindset and knowledge	–	Capitalize user's knowledge for innovation (idea generation, prototyping) [P8] Cumulative changes have a positive effect on learning and success rate [P42]	Formalizing activities forces implicit understandings become explicit [P17] Generating and transferring of insights is key for reuse, e.g., business model cockpit [P54]	Generating and transferring of insights is essential for reuse [P54]

folio without a matching increase in revenue. They propose an iterative learning process based on an integrated business model design and engineering using System Dynamics (SD). SD is used to specify the business models run-time behavior over time, but they conclude that the provision and further development of this approach are crucial in further studies. Kindström identifies vital aspects in **define**, **execute** and **governance** when changing into a service-based business model, and also recognizes the challenge of staying profitable [P24]. However, he makes no specific contribution how to improve efficiency or effectiveness and concludes that more research is needed to link a transition to profitability and growth.

To enhance the effectiveness of collaborative networked organizations, Romero & Molina propose an experience-centric network reference framework based on open-business models (co-innovation/open innovation) [P8]. By integrating a multi-value perspective with a multi-stakeholder approach, one can capitalize on the networked organization's knowledge to achieve better business models (e.g., better risk management and transparency through value co-creation). They present no evidence for improved effectiveness or efficiency.

Partial pattern category papers [P32, P37, P42, P56] focus on the **define** process in combination with **governance** to ensure the expected results. Osterwalder et al. discuss how a formalized model can help to react to external events with speed and effectiveness, but presents no empirical evidence thereof [P32]. Salgado et al. argue that the gap in the business-IS/IT dialogue, which in turn leads to inefficient and non-effective IS/IT solutions, partly comes from: 1) the lack of formality; and 2) high dependency on specific and skilled analysts, when deriving IS/IT requirements from business goals [P56]. They propose the use of PGR (process-level use cases, goals, and rules) to improve traceability and the alignment of Business and IS/IT as a way to improve effectiveness (of both developing and running the IS/IT solution). To close the gap in the business-IS/IT dialog and increase efficiency, they propose a method how to generate a BMC from goals and rules to improve decision making and increase traceability. The method has only been tested on a small, manual scale with considerable limitations: 1) a high dependency on individual analysts and their knowledge and business heuristics; and 2) limited scope due to the amount of human resources needed. Conclusions on effectiveness and efficiency for their work are too early to derive. Ghezzi discusses business strategy under discontinuity and presents three tools to help managers identify a signaling "vector of inputs" to trigger a strategic re-planning process [P37]. He refers to the relation between the business model performance and a company's performance but makes no claims on effectiveness or efficiency with his contribution. Andries & Debackere instead look at the **define** and **execute** processes in their discussion how adaptation and performance are related to new technology-based businesses [P42]. They conclude that business model adaptation is beneficial in less mature, capital-intensive and high-velocity businesses, as it reduces failure rates in dependent business units. However, they do not detail how this can be done using BM.

The *Single pattern* category includes studies [P17, P35, P45] focusing on the **define** process and advocates more research addressing effectiveness and efficiency. Hacklin & Wallnöfer discuss how the business model is applied for strategic decision making [P17]. They explore implications and limitations of using a business model as a "strategizing device" and how BM is forcing to formalize current activities and make implicit understandings. They propose future research on the effectiveness of business: 1) deal with technical aspects how to systematically use BM to improve effectiveness; 2) to test the linguistic legitimacy of various frameworks for BM; and 3) improve the effectiveness of different representational modes of the business model to gain pragmatic validity. Buder & Felden evaluate the efficiency of representation and formalization of value models (e^3value and REA) to understand business models [P35]. They discuss the impact of business processes on value creation and stress the importance of consistency between business and process modeling. They find e^3value

to be more effective and efficient in improving the linkage between BM and business processes. Doganova & Eyquem-Renault investigate the commercialization of technology in the first years of new ventures and the dual role the business model play [P45]. They argue the "performative" role as a demonstration and as a scale model that gradually bring the company's business into existence. They also conclude that empirical findings still fail to provide convergent results regarding the effectiveness of business models.

To summarize, the improvements associated with efficiency and effectiveness are neither substantiated by empirical evidence nor grounded in empirical data. Given the diverse contextual settings in the studies and the dependence of the BM approach, it remains an open question whether the application of any of the identified practices results in increased or decreased efficiency or effectiveness for a company's business. Any outcome variations may simply be a result of fluctuating contextual or environmental factors rather than the application of a BM method or technique. Reaching reasonable coverage of efficiency and effectiveness as external factors require considering several measurable internal factors. With a reasonable coverage of relevant internal factors and taking into account contextual factors, we most likely operate on tens of independent variables that need precise definition and measurement instruments. Given this, we argue that none of the identified studies come near to the required level of details to be able to consider their measurements trustful (except for Andries & Debackere linking business model adaptation to a company's performance via a survival rate measurement and other variables collected from the annual CorpTech directory [P42]).

We concur with Zott et al. that *literature is developing largely in silos, according to the phenomena of interest to the respective researcher* [12]. We conclude that business model research still lacks a consolidated view of what a business model is, while at the same time being forced to address more complexity (e.g., dynamic business models, co-creation, collaboration, and ecosystems with a growing number of stakeholders).

6. Research synthesis

6.1. An analysis of business modeling trends

We synthesized five main trends within our surveyed literature on BM:

- Business models as the building blocks, and the structure of a business model construct as a cornerstone for analyzing, planning and managing competitive and strategic advantages [P1, P2, P3, P4, P9, P13, P16, P19, P29, P32, P40, P41, P51]. Much research is put into frameworks, methods, and tools but the effectiveness and efficiency when integrating this research into practical solutions still miss empirical evidence.
- Locus of the company is shifting to the ecosystem resulting in an explosion of new roles and values that need consideration, as they are connected to the value creation/capture logic [P2, P3, P4, P6, P21, P53, P57]. This trend makes future research more complicated and time consuming, given the lack of consolidated body knowledge on what a business model is and how it can be represented to support experimentation and efficient information management.
- Experimentation and operationalization of flexible business models, to manage the speed of change fueled by technology innovation and the digitalization of the value delivery [P1, P2, P9, P13, P15, P18, P49, P51]. We too, argue for a more cross-disciplinary agenda [57], as business modeling is facing the same challenges as agile requirement engineering and software development has been looking at for the past 10 years trying to increase speed and productivity [63].
- Changeability and modularity as ways to strategically address all new roles and values via choices to enable faster transitions from strategy to execution (operationalization) [P1, P3, P5, P6, P23, P25, P26, P27]. By systematically approaching the information management related to business models, changeability, and modularity, parts of the

practices for business modeling may become automated as a solution to faster transitions.
– A growing need for multifaceted optimization of business models, as fueled by new roles and new values, as a contrast to the currently more dominant single dimension of cost and revenue [P2, P7, P8, P9, P26, P53], often leading to sub-optimal solutions. Such optimization will drive a need for more sophisticated decision support and higher levels of automation in the governance of business models and business model execution.

We found no solutions or evidence related to multifaceted optimization of business models, while at the same time multiple studies highlighted the need for alignment of strategy and execution (daily operations). In combination with the two related trends of experimentation and changeability, we identified a common denominator in governance, as a foundation for faster and more transparent decision-support (for all roles in their interactions). Also, we found no systematic mechanism for organizational learning that potentially could minimize misunderstandings and improve decisions, even though organizational learning is important for successful BM [P9, P46].

We believe an important step towards such multifaceted optimization of business models lies in understanding how the business modeling practice connects to governance for evaluating effectiveness and efficiency of a company. We, therefore, propose CGM to facilitate the exploration of a governance framework for evaluating effectiveness (creating the right values) and efficiency (while using a minimum of resources).

6.2. A conceptual governance model (CGM) for exploring governance and evaluating effectiveness and efficiency of BM

We synthesized CGM for exploring governance and evaluating effectiveness and efficiency of BM. CGM is presented in Figure 5 and is inspired by Zott and Amit's work on business models as activity systems that create value in transactions [5], and influenced by the theories of learning and knowledge creation by Pask and Nonaka [20,64]. CGM links governance to BM via the antecedents (H1, H2), the business model (H3), real-world interactions (creating value and learning), and consequences (H4) as defined by Höflinger [7]. It is a conceptualization of the diversity of the problem of BM concerning value, effectiveness, and efficiency. We propose CGM be used for exploring experimentation in business modeling and designing a scalable IT solution. We believe the concept of "context frame" and intent-driven systems [65] offers an exciting path forward and will be elaborated as part of our future work.

Figure 5 illustrates how the BM practice facilitates experimentation with a business model through a set of interactions between actors involved in the *define* (P_0) and *execute* (P_1) processes. P_0 and P_1 are abstracted from the underlying phases of interaction and learning, as mentioned both by Nonaka (dialogue vs. practice) and Pask (explaining vs. demonstrated understanding). The processes exist in a context, influencing and influenced by the environment on different abstraction levels (and each process can also be seen as a representation of an activity system with its interdependent activities in line with Zott and Amit's work). Please note that both processes are highly context-specific, but always executed in pairs (as interactions of activity systems), e.g., context A = producing a strategy, context B = translating the same strategy into an operationalized business model in products. Therefore, P_0 and P_1 interact in a highly recursive, non-linear, interactive manner.

Depending on the context, different tasks and activities are executed (by sharing and modifying information related to various parts of the company's strategies, organizations, policies, rules, and products in close relation to the ecosystem). Such context dependency is a critical and challenging factor for a process-centric implementation of activities since reuse easily becomes complex, unpredictive, and slow [66].

Governance is an abstraction of goals, measurements, follow-up, rules, knowledge, and insights. Relationships r_1 and r_2 represent the relationship between governance of *define* and *execute* processes and how governance is used

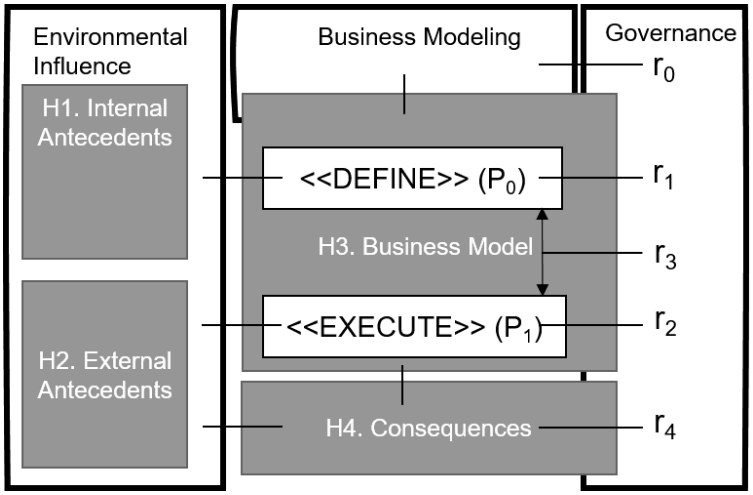


Figure 5. CGM, a conceptual governance model for exploring effectiveness and efficiency in relation to BM with key relationships r_x

to form an agreement (alignment of strategy and execution via goals, objectives, rules, measurements, and knowledge). r_3 represents the relationship between the *define* and *execute* processes and how governance is involved in tracking daily progress and facilitating alignment including change management (by executing in relation to the agreements established/updated via r_1 and r_2). r_0 is used to manage the effectiveness and efficiency of the BM process, while r_4 is used to control the outcome of the business (model execution). Our future work aims to develop these relationships into software interface in accordance with intent-driven systems [65].

Sustaining competitive advantage requires constant change [4]. Fundamental to this change is to understand the difference (make an assessment) between the current position (means) and the desired position (ends). Successful change is thus a multifaceted function of alignment between ends and means, maintained by timely actions to modify ends and the means in response to the environmental influences and consequences. The purpose of the relationships r_0–r_4 in Figure 5 is to manage successful change systematically. However, common to all studies (with quotes of purpose, Appendix D) is a lack of details describing the r_1–r_3 relationships and how the alignment can be achieved and maintained.

The importance of aligning the execution with the strategy is specifically addressed by papers [P6, P32, P59] (without empirical results). Only Salgado et al. suggest solutions to how that could be done (methods and representation of information) [P59]. Ballon proposes an analytical framework and discuss how BM is interpreted as (re)configuration of control parameters (combination of assets, vertical integration, customer ownership, modularity, distribution of intelligence, interoperability) and value parameters (cost sharing model, revenue model, revenue sharing model, positioning, customer involvement, intended value) [P6]. Osterwalder et al. advocate formalization of business models to create traceability between business (the building plan) and execution (IS/IT systems) [P32]. Giessmann et al. extend Osterwalder et al.'s propositions to build a model that can analyze and compare business models, but their work does not address the issues of aligning and daily execution of a business model [P55]. Salgado et al. also build on Osterwalder's BMC and discuss how to generate a BMC from business goals, rules, and processes, but do not further connect the results to the IS/IT realization and daily operations [P56]. They also discuss the alignment between business and IS/IT from the lens of business model artifacts, strategy and goal modeling, as well as enterprise modeling [P59]. They formulate the primary challenge as *Achieving alignment per se is not enough, organizations have to reach the alignment state and maintain it alongside its evolution.*

The quotes for challenges and benefits (Appendix D) also lack details describing the relationships r_1–r_3 in Figure 5. Also, there are 62% more quotes than for purposes, which could be explained by that benefits and challenges are often more specific by nature than the corresponding purposes. The identified quotes indicate a more inhomogeneous nature regarding contextual settings, resulting in a scattered picture of benefits and challenges. We speculate this is a result of each paper framing their conclusions with some form of benefits or challenges, rather than constructing them from empirical findings.

The papers within the governance column and assessment row (see Appendix D) present important aspects of goals, rules, measurements, options, flexibility, and knowledge. However, they do not propose solutions on how these concepts (with artifacts) should be represented or managed to create traceability to, and alignment with, the define and execute processes (via r_1, r_2, r_3) in Figure 5.

Six papers [P2, P22, P29, P32, P36, P54] cover all three columns (define, execute, and governance), but no author elaborates on the relationships r_1–r_3 (alignment of define and execute processes using governance), see Table 3. Rohrbeck et al. study eight companies and discuss how collaborative BM can improve both define and execute processes [P2]. They report improvements in four areas (dealing with uncertainty, finding creative solutions, facilitating a strategic discussion, and allowed to start the innovation planning), but provide little details or empirical evidence as to how well it works. Baden-Fuller & Morgan scan the literature and discuss business models as models, describing their multivalent character and the wide range of usage [P22]. They conclude *Business models are not recipes or scientific models or scale and role models [. . .] they play any – or all – these roles, often at the same time.* Osterwalder et al. propose eight propositions for BM that need to be tested [P32]. Zott et al. in their review six years later reveal that scholars still do not agree and that literature is developing in silos [P29]. Cortimiglia et al. explore, in a large empirical-based investigation, the relationship between the strategy making process and business model innovation (BMI) [P36]. They summarize a large number of purposes found in literature, which also matches the improvement areas we have identified, see section 5.1. Their findings validate the role of business model innovation as a valuable tool for, and link, between strategy execution and operationalization. Meier & Bosslau, in their case study, propose an integrated design and engineering approach as an iterative learning process based on system dynamics. They conclude that further development of modeling and simulation that depicts the dynamics and flexibility in the whole life-cycle is one of the key challenges for business model research (in a context of industrial product service systems) [P54].

7. Implications for research and practitioners

The results suggest that business model (and BM) is a diverse research area which would benefit from more aggregation efforts [P29, P40, P3, P9] on how business models could address the vast set of purposes and practices for BM, and what effects BM have on effectiveness and efficiency of a company. More work is needed to consolidate these different angles of the business model construct into a scalable, practically useful representations that will facilitate innovation, experimentation, and operationalization of the business model. The lack of coherence is more recently investigated by Massa et al. [67], as they identify possible reasons for the current lack of agreement in literature as terms and concepts slowly morph over time.

In the same vein (seen from a practitioners' side), Gartner[7] in 2014 points out that *digital business should not be considered an IT program and should instead become an enterprise mindset and lingua franca, with digital expertise spread across the enterprise and value ecosystem.*

Our results confirms the above and highlight a challenging issue for effectively and efficiently

[7]Gartner identifies six key steps to build a successful digital business, https://www.gartner.com/newsroom/id/2745517

defining contexts to improve understanding and communication in BM literature. We also note a potentially strong correlation between flexibility, effectiveness, and efficiency (all 13 papers addressing RQ2 also address aspects of flexibility and variability in the realization, IC2 and IC3, see Table 2).

We recommend the following topics to be added to a cross-disciplinary agenda for BM:

- Further exploring how contextual information in the business model construct could be systematically represented, structured, and stored. The improved representation of contextual information is going to increase effectiveness and efficiency when creating, modifying, and deleting information needed to transform strategies into tactics and daily execution, e.g., facilitating business model choices, including a residual set of choices related to tactics, and deciding on choices controlling daily interactions between stakeholders (as controlled by a set of configuration parameters and rules in software applications). A business model construct must support collaborative and role-based interaction, including exchange and interpretation of contextual information, scalable to thousands of actors, and across corporate borders. We believe intent-driven systems [65] could be a way forward for this purpose.
- Connecting the BM practice with Learning Theory would help to create a model that can help explain: 1) how value creation and stakeholder motivation is derived from, and connected to, daily interactions; 2) how daily interactions, in combination with organizational learning, shape the transformation of strategy into execution; and 3) how organizational learning influences the process of BM. These aspects become increasingly important since experimentation with value co-creation and business models are gaining interests [P2, P9, P13, P18]. This implies BM to be involved, not only in strategy and planning but also in the operationalization and follow-up of the business model, as the focus of a business model is shifting beyond the company borders into the ecosystem.

The implications for industry originate mainly from the lack of tangible results linking efficient BM to efficient and effective businesses. We recommend managers to investigate and build awareness of the following aspects:

- Systematically converting experience into knowledge will help the organization identifying and verbalizing (new) values and motivators relevant to the business. Investigate how to incorporate organizational learning (OL) [68] into everyday practices and business processes to support experimentation with business models, e.g., what is the current level of OL? How is OL incorporated into important business processes? Which roles are currently not involved in structured OL? How is OL related to the fulfillment of goals, an organization's creativity and motivation, and incentives?
- Critical components in any SIPD business model are concepts such as value co-creation, collaborative value networks, and acquiring resources beyond the control of the company (i.e., creating an ecosystem of partners and customers). How to prepare a company's staff and products to these concepts? How do you facilitate similar activities for your partners? These ideas will affect the products and offerings but also fundamentally change most aspects of a company's policies and business processes including incentive structures and management systems (e.g., sharing of information internally/externally and risk management). We believe the introduction of a value vocabulary, to facilitate more precise understanding and definitions of business-critical concepts, is a concrete and valuable first step, e.g., SVM [48].
- What factors hinder business model experimentation? What level of business flexibility is required (and used)? How is that flexibility implemented in the products, organization, business processes, and management systems? The value creation process is highly interdependent and not well suited for isolated practices [P14, P15, P30]. Business modeling

could become a tool to bridge these practices [P2] and SIPD companies should not see software architectures and methods as costs. It's a significant investment that facilitates experimentation while adding to the value creation. Such investments in business flexibility will become a crucial source of innovation and an enabler for automating business processes, resulting in an increased efficiency and competitive advantage.

– A governance mechanism is a critical element to build a commitment to experimentation and the development of the appropriate business flexibility. The mechanism should support multi-contextual governance views, maintaining traceability between all choices (strategical, tactical, and operational) and the views must be based on data from different contextual situations (narrative, planning, development, daily operational tasks, phase out, etc.) [65].

8. Conclusions

This systematic literature review explores the purpose of business modeling and its impact on effectiveness and efficiency of a company's business. Most companies invest in business modeling, but remain uncertain whether their investments allow them to change and adapt their business fast enough.

Our results show that the reported benefits are unsubstantiated or claimed with limited empirical evidence and the challenges are dispersed. The most common challenge is how to deal with the dynamics of business models, and most of the quotes on challenges relate to the non-existing solutions for governance (representation, simulation, decision-support, and feedback) of the proposed frameworks and methods.

The improvements associated with efficiency and effectiveness of BM are neither substantiated by empirical evidence nor grounded in empirical data. Given the diverse contextual settings in the studies and the dependence of the BM approach, it remains an open question whether the application of any of the identified practices results in increased or decreased efficiency or effectiveness for a company's business. Any outcome variations may simply be a result of fluctuating contextual or environmental factors rather than the application of a BM method or technique.

We concur with Zott et al. that *literature is developing largely in silos, according to the phenomena of interest to the respective researcher* [12]. Since the influential work by Osterwalder et al. on business models [9], which later gained a lot of interest among practitioners[8], researchers are still reporting that business models and BM is a diverse research area missing an agreed definition of business model. It is an area that would benefit from more aggregated cross-disciplinary research results [57, 67].

Supported by our results, we argue that:

– Related to RQ1, what makes business model research results challenging to analyze, compare, and combine is the lack of a systematic approach in describing the contextual information used to define the context for a specific business model construct and business modeling practice. The lack of systematic contextual information leads to inefficient communication, knowledge creation, and organizational learning, which affects the quality of decisions (on all levels). A consequence for business modeling is misalignment between the business model and its realization, which negatively affects the value creation (effectiveness) and the efficiency. By improving the information management parts of these processes, tasks may become automated, opening up for new ways of specifying and visualizing strategies, goals, and operational consequences, as related to effectiveness and efficiency.

– Related to RQ2, we conclude that governance is going to gain importance, as it must effectively support a chain of continuous adaptations and learning (experimenting). Such governance can enforce a continuous (business model) design aligned with the continuous

[8] Originally called the Business Model Generator in 2010, now changed into a commercial product https://strategyzer.com/canvas.

(business model) execution. We further argue that governance is the primary challenge for business modeling, and that (continuous) business modeling can be used (via governance) to effectively and efficiently cope with change, by connecting the definition of strategy to the execution of operations in daily decisions and activities as depicted in Figure 5.

– By combining above conclusions, that the lack of a rigorous, scalable, context-dependent (software and IT) representation of the business model, in combination with efficient governance mechanisms (to manage needed flexibility), are currently significant obstacles for progressing the research area and supporting the industry in managing innovation in co-creation-driven (software-intensive) business ecosystems.

We, therefore, believe our conceptual governance model is a significant step to explore and identify how the business modeling practice could become an integrated cornerstone in a more effective and efficient software-intensive product development enterprise. Our conceptual governance model can facilitate the creation a common business model construct including mechanisms to support effective and efficient governance with value-based decision-support for all affected roles and stakeholders.

Also, we believe our extensive, cross-disciplinary review of the business model literature, seen from the perspective of software and software-intensive products, is a valuable contribution for the Software Engineering community when trying to address the digitalization's effects on software engineering and software product development.

Our next steps in our research towards efficient and effective business modeling are to use our proposed conceptual model to identify essential characteristics of a governance framework and a scalable business model construct, as required to facilitates effective and efficient operationalization of a business model. We will also verify the conceptual model with practitioners to ensure that our results can be disseminated by industry.

Acknowledgment

We are grateful for the constructive and helpful comments on early drafts received from Prof. Lars Bengtsson, LTH, Sweden. This work has been supported by the Professional Licentiate of Engineering (PLEng) Pilot Run 2014–2018 in cooperation with Ericsson AB. This work is also supported by the IKNOWDM project (20150033) from the Knowledge Foundation in Sweden.

References

[1] C. Matt, T. Hess, and A. Benlian, "Digital transformation strategies," *Business and Information Systems Engineering*, Vol. 57, No. 5, 2015, pp. 339–343.

[2] A. Bharadwaj, O. El Sawy, P. Pavlou, and N. Venkatraman, "Digital business strategy: Toward a next generation of insights," *MIS Quarterly*, Vol. 37, No. 2, 2013, pp. 471–482.

[3] A. Osterwalder and Y. Pigneur, *Business model generation: A handbook for visionaries, game changers, and challengers*, 2010.

[4] R. Casadesus-Masanell and J.E. Ricart, "From strategy to business models and onto tactics," *Long Range Planning*, Vol. 43, No. 2–3, 2010, pp. 195–215.

[5] C. Zott and R. Amit, "Business Model Design: An activity system perspective," *Long Range Planning*, Vol. 43, No. 2–3, 2010, pp. 216–226.

[6] M. Eurich, T. Weiblen, and P. Breitenmoser, "A six-step approach to business model innovation," *International Journal of Entrepreneurship and Innovation Management*, Vol. 18, No. 4, 2014, pp. 330–348.

[7] N.F. Höflinger, "The business model concept and its antecedents and consequences – Towards a common understanding," *Academy of Management Proceedings: Organization Development & Change*, Vol. 2014:1, 2014.

[8] D.P. Lepak, K.G. Smith, and M.S. Taylor, "Introduction to special topic forum value creation and value capture: A multilevel perspective," *Academy of Management Review*, Vol. 32, No. 1, 2007, pp. 180–194.

[9] A. Osterwalder, Y. Pigneur, and C.L. Tucci, "Clarifying business models: Origins, present, and future of the concept," *Communications of the Association for Information Systems*, Vol. 15, No. 1, 2005, pp. 1–25.

[10] C. Baden-Fuller and S. Haefliger, "Business Models and Technological Innovation," *Long Range Planning*, Vol. 46, No. 6, 2013, pp. 419–426.

[11] J. Krumeich, D. Werth, T. Burkhart, and P. Loos, "Towards a component-based description of business models: A state-of-the-art analysis," in *18th Americas Conference on Information Systems 2012, AMCIS 2012*, Vol. 1, 2012, pp. 266–277.

[12] C. Zott, R. Amit, and L. Massa, "The business model: Recent developments and future research," *Journal of Management*, Vol. 37, No. 4, 2011, pp. 1019–1042.

[13] P. Zave, "Classification of research efforts in requirements engineering," *ACM Computing Surveys (CSUR)*, Vol. 29, No. 4, 1997, pp. 315–321.

[14] E. Kavakli, "Goal-oriented requirements engineering: A unifying framework," *Requirements Engineering*, Vol. 6, No. 4, 2002, pp. 237–251.

[15] B. Ramesh, L. Cao, and R. Baskerville, "Agile requirements engineering practices and challenges: an empirical study," *Information Systems Journal*, Vol. 20, No. 5, 2010, pp. 449–480.

[16] J. Bosch and P. Bosch-Sijtsema, "From integration to composition: On the impact of software product lines, global development and ecosystems," *Journal of Systems and Software*, 2010.

[17] J. Buder and C. Felden, "Evaluating business models: Evidence on user understanding and impact to BPM correspondence," *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2012, pp. 4336–4345.

[18] W. Zheng, B. Yang, and G.N. McLean, "Linking organizational culture, structure, strategy, and organizational effectiveness: Mediating role of knowledge management," *Journal of Business Research*, Vol. 63, No. 7, 2010, pp. 763–771.

[19] E. Frökjær, M. Hertzum, and K. Hornbæk, "Measuring usability," in *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '00*. ACM Press, 2000, pp. 345–352.

[20] G. Pask, *Conversation Theory – Applications in Education and Epistemology*. Amsterdam and New York: Elsevier Inc., 1976.

[21] K. Petersen and C. Wohlin, "Context in industrial software engineering research," in *3rd International Symposium on Empirical Software Engineering and Measurement, ESEM*, 2009, pp. 401–404.

[22] T. Haaker, H. Bouwman, and E. Faber, "Customer and network value of mobile services: Balancing requirements and strategic interests," in *ICIS 2004 Proceedings. Paper 1*, 2004.

[23] M.M. Al-Debei and G. Fitzgerald, "The design and engineering of mobile data services: Developing an ontology based on business model thinking," in *IFIP Advances in Information and Communication Technology*, Vol. 318, 2010, pp. 28–51.

[24] S. Jansen, "Measuring the health of open source software ecosystems: Beyond the scope of project health," *Information and Software Technology*, Vol. 56, No. 11, 2014, pp. 1508–1519.

[25] M. Page and L.F. Spira, "Corporate governance as custodianship of the business model," *Journal of Management & Governance*, Vol. 20, No. 2, 2016, pp. 213–228.

[26] W. Reim, V. Parida, and D. Örtqvist, "Strategy, business models or tactics – What is product-service systems (PSS) literature talking about?" in *Proceedings of the International Conference on Engineering Design, ICED*, Vol. 4, 2013, pp. 309–318.

[27] C. Baden-Fuller and M.S. Morgan, "Business models as models," *Long Range Planning*, Vol. 43, No. 2–3, 2010, pp. 156–171.

[28] R. Rohrbeck, L. Konnertz, and S. Knab, "Collaborative business modelling for systemic and sustainability innovations," *International Journal of Technology Management*, Vol. 63, No. 1/2, 2013, p. 4.

[29] L. Doganova and M. Eyquem-Renault, "What do business models do? Innovation devices in technology entrepreneurship," *Research Policy*, Vol. 38, No. 10, 2009, pp. 1559–1570.

[30] H. Chesbrough, "Business Model Innovation: Opportunities and Barriers," *Long Range Planning*, Vol. 43, No. 2–3, 2010, pp. 354–363.

[31] R.G. McGrath, "Business Models: A Discovery Driven Approach," *Long Range Planning*, Vol. 43, No. 2–3, 2010, pp. 247–261.

[32] M. Sosna, R.N. Trevinyo-Rodríguez, and S.R. Velamuri, "Business Model Innovation through Trial-and-Error Learning," *Long Range Planning*, Vol. 43, No. 2–3, 2010, pp. 383–407.

[33] P. Ballon, "Business modelling revisited: the configuration of control and value," *Info*, Vol. 9, No. 5, 2007, pp. 6–19.

[34] C.E. Salgado, R.J. Machado, and R.S. Maciel, "An OMG-based meta-framework for alignment of IS/IT architecture with business models," in *9th International Conference on the Quality of Information and Communications Technology*, 2014.

[35] Y.L. Doz and M. Kosonen, "Embedding strategic agility: A leadership agenda for accelerating business model renewal," *Long Range Planning*, Vol. 43, No. 2–3, 2010, pp. 370–382.

[36] S. Schneider and P.A.T. Spieth, "Business model innovation and strategic flexibility: insights from

an experimental research design," *International Journal of Innovation Management*, Vol. 18, No. 6, 2014, pp. 1–22.

[37] A. Richter, T. Sadek, and M. Steven, "Flexibility in industrial product-service systems and use-oriented business models," *CIRP Journal of Manufacturing Science and Technology*, Vol. 3, No. 2, 2010, pp. 128–134.

[38] J. Moore, "The rise of a new corporate form," *Washington Quarterly*, Vol. 21, No. 1, 1998, pp. 167–181.

[39] R. Normann and R. Ramirez, "From value chain to value constellation: Designing interactive strategy," *Harvard Business Review*, Vol. 71, No. 4, 1993, pp. 65–77.

[40] T. Berger, R.H. Pfeiffer, R. Tartler, S. Dienst, K. Czarnecki, A. Wasowski, and S. She, "Variability mechanisms in software ecosystems," *Information and Software Technology*, Vol. 56, 2014, pp. 1520–1535.

[41] M. Schief and P. Buxmann, "Business Models in the Software Industry," in *45th Hawaii International Conference on System Sciences*, 2012, pp. 3328–3337.

[42] R. Casadesus-Masanell and G. Llanes, "Mixed source," *Management Science*, Vol. 57, No. 7, 2011, pp. 1212–1230.

[43] V. Sambamurthy, A. Bharadwaj, and V. Grover, "Shaping agility through digital options: Reconceptualizing the role of information technology in contemporary firms," *MIS Quarterly: Management Information Systems*, Vol. 27, No. 2, 2003, pp. 237–264.

[44] A. Zolnowski and T. Böhmann, "Business modeling for services: Current state and research perspectives," in *AMCIS 2011 Proceedings*, 2011. [Online]. http://aisel.aisnet.org/amcis2011_submissions/394/

[45] H. Meier, R. Roy, and G. Seliger, "Industrial product-service systems – IPS2," *CIRP Annals*, Vol. 59, No. 2, 2010, pp. 607–627.

[46] H. Meier and M. Boßlau, "Design and engineering of dynamic business models for industrial product-service systems," in *The Philosopher's Stone for Sustainability*, Y. Shimomura and K. Kimita, Eds., 2012.

[47] J. Björkdahl, "Technology cross-fertilization and the business model: The case of integrating ICTs in mechanical engineering products," *Research Policy*, Vol. 38, No. 9, 2009, pp. 1468–1477.

[48] M. Khurum, T. Gorschek, and M. Wilson, "The software value map – An exhaustive collection of value aspects for the development of software intensive products," *Journal of software: Evolution and Process*, Vol. 25, No. 7, 2013, pp. 711–741.

[49] S.C. Lambert and R.A. Davidson, "Applications of the business model in studies of enterprise success, innovation and classification: An analysis of empirical research from 1996 to 2010," *European Management Journal*, Vol. 31, No. 6, 2013, pp. 668–681.

[50] D.J. Teece, "Business models, business strategy and innovation," *Long Range Planning*, Vol. 43, No. 2–3, 2010, pp. 172–194.

[51] M. Morris, M. Schindehutte, and J. Allen, "The entrepreneur's business model: Toward a unified perspective," *Journal of Business Research*, Vol. 58, No. 6, 2005, pp. 726–735.

[52] A. Afuah, *Business Models: A Strategic Management Approach*, 1st ed. New York: McGraw-Hill, 2004.

[53] A. Afuah and C.L. Tucci, *Internet Business Models and Strategies: Text and Cases*. New York: McGraw Hill Higher Education, 2002.

[54] F. Hacklin and M. Wallnöfer, "The business model in the practice of strategic decision making: insights from a case study," *Management Decision*, Vol. 50, No. 2, 2012, pp. 166–188.

[55] C. Zott and R. Amit, "Business model design and the performance of entrepreneurial firms," *Organization Science*, Vol. 18, No. 2, 2007, pp. 181–199.

[56] C. Zott and R. Amit, "The fit between product market strategy and business model: Implications for firm performance," *Strategic Management Journal*, Vol. 29, No. 1, 2008, pp. 1–26.

[57] A. Osterwalder and Y. Pigneur, "Designing business models and similar strategic objects: The contribution of IS," *Journal of the Association of Information Systems*, Vol. 14, No. 5, 2013, pp. 237–244.

[58] A. Giessmann, A. Fritz, S. Caton, and C. Legner, "A method for simulating cloud business models: A case study on Platform as a Service," in *21st European Conference on Information Systems, Completed Research 42*, 2013, pp. 1–12.

[59] C.E. Salgado, J. Teixeira, R.J. Machado, and R.S.P. Maciel, "Generating a business model canvas through elicitation of business goals and rules from process-level use cases," in *Proceedings of the 13th International Conference on Business Informatics Research*, 2014, pp. 1–15.

[60] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering – EASE*, 2014, pp. 1–10.

[61] J. Corbin and A. Strauss, *Basics of Qualitative Research: Techniques and Procedures for Devel-*

oping Grounded Theory, 4th ed. SAGE Publications, Inc., 2015.

[62] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, Vol. 14, No. 2, 2009, pp. 131–164.

[63] I. Inayat, S.S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Computers in Human Behavior*, Vol. 51, 2014, pp. 915–929.

[64] I. Nonaka, R. Toyama, and N. Konno, "SECI, ba and leadership: A unified model of dynamic knowledge creation," *Long Range Planning*, Vol. 33, No. 1, 2000, pp. 5–34.

[65] J. Silvander, M. Wilson, K. Wnuk, and M. Svahnberg, "Supporting continuous changes to business intents," *International Journal of Software Engineering and Knowledge Engineering*, Vol. 27, No. 8, 2017, pp. 1167–1198.

[66] A. Koschmider, M. Fellman, A. Schoknecht, and A. Oberweis, "Analysis of process model reuse: Where are we now, where should we go from here?" *Decision Support Systems*, Vol. 66, 2014, pp. 9–19.

[67] L. Massa, C.L. Tucci, and A. Afuah, "A critical assessment of business model research," *Academy of Management Annals*, Vol. 11, No. 1, 2016.

[68] L. Argote, "Organizational learning: From experience to knowledge," *Organization science*, Vol. 22, No. 5, 2011, pp. 1123–1137.

[69] C.J. Woodard, N. Ramasubbu, F.T. Tschang, and V. Sambamurthy, "Design capital and design moves: The logic of digital business strategy," *MIS Quarterly: Management Information Systems*, Vol. 37, No. 2, 2013, pp. 537–564.

[70] R. Hackney, J. Burn, and A. Salazar, "Strategies for value creation in electronic markets: Towards a framework for managing evolutionary change," *The Journal of Strategic Information Systems*, Vol. 13, No. 2, 2004, pp. 91–103.

[71] E.K. Chew, "Linking a service innovation-based framework to business model design," in *16th Conference on Business Informatics*, Vol. 1. IEEE, 2014, pp. 191–198.

[72] L. Loss and S. Crave, "Agile Business Models: An approach to support collaborative networks," *Production Planning & Control*, Vol. 22, No. 5–6, 2011, pp. 571–580.

[73] D. Romero and A. Molina, "Collaborative networked organisations and customer communities: Value co-creation and co-innovation in the networking era," *Production Planning & Control*, Vol. 22, No. 5–6, 2011, pp. 447–472.

[74] A. Goel, H. Schmidt, and D. Gilbert, "Towards formalizing Virtual Enterprise Architecture," *13th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW)*, 2009, pp. 238–242.

[75] B. Demil and X. Lecocq, "Business model evolution: In search of dynamic consistency," *Long Range Planning*, Vol. 43, No. 2–3, 2010, pp. 227–246.

[76] M. Dubosson-Torbay, A. Osterwalder, and Y. Pigneur, "E-business model design, classification, and measurements," *Thunderbird International Business Review*, Vol. 44, No. 1, 2002, pp. 5–23.

[77] J. Richardson, "The business model: an integrative framework for strategy execution," *Strategic Change*, Vol. 17, No. 5–6, 2008, pp. 133–144.

[78] K. Storbacka and S. Nenonen, "Scripting markets: From value propositions to market propositions," *Industrial Marketing Management*, Vol. 40, No. 2, 2011, pp. 255–266.

[79] J. Gao, Y. Yao, V.C.Y. Zhu, L. Sun, and L. Lin, "Service-oriented manufacturing: A new product pattern and manufacturing paradigm," *Journal of Intelligent Manufacturing*, Vol. 22, No. 3, 2009, pp. 435–446.

[80] D. Kindström, "Towards a service-based business model – Key aspects for future competitive advantage," *European Management Journal*, Vol. 28, No. 6, 2010, pp. 479–490.

[81] H. Meier and W. Massberg, "Life cycle-based service design for innovative business models," *CIRP Annals*, Vol. 53, No. 1, 2004, pp. 393–396.

[82] G. Schuh, W. Boos, and S. Kozielski, "Life cycle cost-orientated service models for tool and die companies," in *Proceedings of the 1st CIRP Industrial Product-Service Systems (IPS2) Conference*, 2009, pp. 249–254.

[83] R. Amit and C. Zott, "Value creation in e-business," *Strategic Management Journal*, Vol. 22, No. 6–7, 2001, pp. 493–520.

[84] H. Bouwman and I. MacInnes, "Dynamic business model framework for value webs," in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS '06)*, 2006.

[85] M.N. Cortimiglia, A. Ghezzi, and A.G. Frank, "Business model innovation and strategy making nexus: evidence from a cross-industry mixed-methods study," *R&D Management*, Vol. 46, No. 3, 2016, pp. 414–432.

[86] A. Ghezzi, "Revisiting business strategy under discontinuity," *Management Decision*, Vol. 51, No. 7, 2013, pp. 1326–1358.

[87] A. Ghezzi, "Emerging business models and strategies for mobile platform providers: A reference framework," *Info*, Vol. 14, No. 5, 2012, pp. 36–56.

[88] P. Andries and K. Debackere, "Adaptation and performance in new businesses: Understanding the moderating effects of independence and industry," *Small Business Economics*, Vol. 29, No. 1–2, 2007, pp. 81–99.

[89] K.J.K. Mason and S. Leek, "Learning to build a supply network: An exploration of dynamic business models," *Journal of Management Studies*, Vol. 45, No. 4, 2008, pp. 774–799.

[90] J. Lindström, "A model for value-based selling: Enabling corporations to transition from products and services towards further complex business models," *Journal of Multi Business Model Innovation and Technology*, Vol. 1, 2014, pp. 67–98.

[91] Y. Ning, H. Fu, and W. Zheng, "Business model dynamics: A case study of Apple Inc." in *18th International Conference on Industrial Engineering and Engineering Management*, 2011, pp. 77–80.

[92] V. Dmitriev, G. Simmons, Y. Truong, M. Palmer, and D. Schneckenberg, "An exploration of business model development in the commercialization of technology innovations." *R&D Management*, Vol. 44, No. 3, 2014, pp. 306–321.

[93] S.W. Short, P. Rana, N.M.P. Bocken, and S. Evans, "Embedding sustainability in business modelling through multi-stakeholder value innovation," in *Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services*, Vol. 397, 2013, pp. 175–183.

[94] Y. Kim, Y. Lee, G. Kong, H. Yun, and S. Chang, "A new framework for designing business models in digital ecosystem," in *2nd International Conference on Digital Ecosystems and Technologies*. IEEE, 2008, pp. 281–287.

[95] K. Mason and S. Mouzas, "Flexible business models," *European Journal of Marketing*, Vol. 46, No. 10, 2012, pp. 1340–1367.

[96] M. Ivarsson and T. Gorschek, "A method for evaluating rigor and industrial relevance of technology evaluations," *Empirical Software Engineering*, Vol. 16, No. 3, 2011, pp. 365–395.

Appendix A. Selected articles

Table A lists all the articles selected through the snowballing methodology. It contains Paper ID, author/bibliographic reference, plus extracted data for rigor and relevance factors (EP3), paper content (EP4), and the number of topics (RQ1+RQ2+IC2+IC3)[9] addressed by the paper. A detailed description of EP3 (including calculation of scores) and EP4 are found in the Appendix C while details of IC1–IC3 are found in Appendix B.

In the main article we use the notation [Paper ID,...] to indicate a reference to one or more of the study's selected papers when we specifically talk about a result or an synthesis thereof. Please note that the start set consists of P1–P10.

Table A. Selected papers including extracted properties

Paper			Rigor (EP3)			Relevance (EP3)				Content	No. of
ID	Authors/Ref	Year	C	SD	V	C	Sc	Su	RM	EP4	RQ+IC
P1	Woodard et al. [69]	2013	1	1	1	1	1	1	1	2	4
P2	Rohrbeck et al. [28]	2013	0.5	1	0	1	0	1	1	1	3
P3	Reim et al. [26]	2013	0.5	0.5	0	0	0	0	0	2	3
P4	Hackney et al. [70]	2004	0.5	0	0	1	1	0	0	3	2
P5	Chew [71]	2014	1	0	0	1	0	0	0	2	4
P6	Ballon [33]	2007	0	0	0	1	0	0	0	1	3
P7	Loss & Crave [72]	2011	0	0	0	1	0	0	0	2	3
P8	Romero & Molina [73]	2011	0	0	0	0	0	0	0	3	3
P9	Höflinger [7]	2014	0.5	1	0	0	0	0	0	2	3
P10	Goel et al. [74]	2009	0.5	0	0.5	0	0	0	0	3	3
P12	Casadesus-Masanell & Ricart [4]	2010	0	0	0	1	0	1	0	3	3
P13	Chesbrough [30]	2010	0	0	0	1	0	0	0	2	3
P14	Demil & Lecocq [75]	2010	1	0	0.5	0	0	1	0	2	2
P15	Doz & Kosonen [35]	2010	0	0	0	0	0	0	0	1	2
P16	Dubosson-Torbay et al. [76]	2002	0	0	0	0	0	0	0	2	2
P17	Hacklin & Wallnöfer [54]	2012	1	0.5	0	1	1	1	1	1	3
P18	McGrath [31]	2010	0	0	0	0	0	0	0	1	3
P19	Richardson [77]	2008	0	0	0	0	0	0	0	2	3
P20	Storbacka & Nenonen [78]	2011	1	1	1	1	1	1	1	2	2
P21	Zott & Amit [5]	2010	0	0	0	1	0	0	0	2	2
P22	Baden-Fuller & Morgan [27]	2010	0	0	0	0	0	0	0	2	3
P23	Gao et al. [79]	2011	0	0	0	1	0	1	0	3	2
P24	Kindström [80]	2010	1	1	0.5	1	1	1	1	2	4
P25	Meier & Massberg [81]	2004	0	0	0	0	0	0	0	3	2
P26	Meier et al. [46]	2010	0	0	0	1	0	1	0	3	3
P27	Richter et al. [37]	2010	0	0	0	0	0	0	0	2	2
P28	Schuh et al. [82]	2009	0	0	0	0	0	0	0	3	1
P29	Zott et al. [12]	2011	0.5	1	1	0	0	0	0	2	4
P30	Amit & Zott [83]	2001	1	1	1	1	1	1	1	3	2
P31	Baden-Fuller & Haefliger [10]	2013	0.5	0	0	0	0	0	0	3	3
P32	Osterwalder et al. [9]	2005	0	0	0	0	0	0	0	2	3
P33	Al-Debei [23]	2010	0.5	0	0	1	1	1	1	2	3
P34	Bouwman [84]	2006	0	0	0	0	0	1	0	3	3
P35	Buder &Felden [17]	2012	1	1	1	0	1	0	0	1	4
P36	Cortimiglia et al. [85]	2015	1	1	1	1	1	1	1	1	2
P37	Ghezzi [86]	2013	0.5	0.5	0.5	1	1	1	1	1	4

[9] IC1–IC3 are topic-oriented while IC4 and IC5 are related to rigor and relevance.

Paper ID	Authors/Ref	Year	Rigor (EP3) C	SD	V	Relevance (EP3) C	Sc	Su	RM	Content EP4	No. of RQ+IC
P38	Ghezzi [87]	2012	0.5	0	0	1	1	1	1	3	2
P39	Haaker et al. [22]	2004	0.5	0	1	1	1	1	1	2	2
P40	Krumeich et al. [11]	2012	0	0.5	0.5	0	0	0	0	2	2
P41	Zolnowski & Böhmann [44]	2011	0.5	0.5	0	0	0	0	0	1	2
P42	Andries & Debackere [88]	2007	1	1	0.5	1	1	1	1	3	2
P43	Björkdahl [47]	2009	0.5	0.5	0.5	1	1	1	1	3	2
P44	Casadesus-Masanell & Llanes [42]	2011	1	0	0	0	0	0	0	3	2
P45	Doganova & Eyquem-Renault [29]	2009	0.5	0.5	0	1	1	1	1	2	4
P46	Mason & Leek [89]	2008	0.5	0.5	0	1	1	1	1	3	2
P48	Lindström [90]	2014	0.5	0.5	1	1	1	1	1	3	2
P49	Eurich et al. [6]	2014	0.5	0.5	0	1	1	1	1	1	3
P50	Ning et al. [91]	2011	0.5	0.5	0	1	1	0	0	3	0
P51	Dmitriev et al. [92]	2014	1	1	1	1	1	1	1	1	2
P52	Schneider & Spieth [36]	2014	0.5	0.5	0.5	1	0	0	1	1	3
P53	Short et al. [93]	2013	0	0	0	1	1	0	1	1	2
P54	Meier & Boßlau [46]	2013	0.5	0	0	1	1	1	1	1	4
P55	Giessmann et al. [58]	2013	0.5	0	0.5	1	0	0	0	3	3
P56	Salgado et al. [59]	2014	0.5	0	0	1	1	1	1	1	3
P57	Kim et al. [94]	2008	1	0	0	0	0	0	0	3	2
P58	Mason & Mouzas [95]	2012	1	1	1	1	1	1	1	3	2
P59	Salgado et al. [34]	2014	0	0	0	0	0	0	1	3	2

Appendix B. Inclusion and exclusion criteria

To identify literature related to our research questions, we developed the Inclusion criteria (IC) and Exclusion criteria (EC) listed in Table B. These criteria allow us to explore why BM is used, how it is applied, and what solutions currently exist. Since our research topic covers multiple research disciplines, we decided to address the RQs by designing the IC as wide as possible, to give us a large variety of articles discussing BM (IC1) in any relationship to effectiveness and efficiency. To evaluate BM efficiency, it is important to connect the business strategy via the business model to the execution of the business model with a traceability to daily operations and results. So to understand if business modeling enables effectiveness and efficiency, we want to know how a business model can be operationalized by developing the right type of flexibility (variability in the realization, IC3) matching all desired strategical and tactical choices (business flexibility, IC2).

Business modeling allows an organization to identify and prioritize changes to current business operations (content, activities, and governance). This change is continuously translated into a realization of the business model, through experimentation or otherwise, by understanding how the desired flexibility can be operationalized using modularity in design and software-based systems to support content, activities (all stakeholders, e.g., internal organization, partners, suppliers, and customers) and governance.

Effectiveness and efficiency should be evaluated from the gap between all strategic and tactical choices, in combination with how the organization (and supporting software) utilize the remaining flexibility to create satisfied customers in everyday transactions. The dilemma of not only implementing the right flexibility (supporting the needed business options) but also implementing it efficiently, is key to success, i.e., the right level of variability in the realization combined with the appropriate changeability in the realization to facilitate experimentation with the operationalized business model.

The selection critera was based on IC1 AND (IC2 OR IC3 OR IC4 OR IC5) to achieve a broad selection of papers as possible. If only the term Business model were used (and not specifically Business modeling), the paper could still be a candidate if it referred to activities related to creating, maintaining, or otherwise using a business model.

Appendix C. Data Extraction properties

Table C lists the data extraction properties used for this study and maps their relevance to each RQ.

Properties EP1-EP4 are evaluated per paper and used to analyze the relevance to industry for each paper's contribution. Properties EP5-EP9 use open coding and the extracted data was thematically and narratively analyzed.

Property EP1 and EP2 are subset of property EP3 (Rigor & Relevance) where property EP2 categories the paper's context. We extend the definition of Context (EP3 [96]), by adding (large-scale) Software intensive industry. The relevance parameter (EP3), we coded with binary weights (originally proposed as plain sum of 0 or 1), allowing us to visualize the impact of different relevance aspects. The weights were guided by RQ1, hence setting our priority: Industry (8), Scale (4), Subjects (2) and Research method (1), e.g. a value of 9 or higher would represent anything in "industry" with at least one additional relevance aspect met. Originally the Relevance element of property EP3 focus on the paper's context in relation to industry so we added property EP4 (Paper content) to map the relevance of each paper's content related to answering the RQs.

EP5 corresponds to our inclusion criteria (IC). EP6 was used to look for patterns on the business model construct as to describe what it is, why it is important and how it is used. This is important since the topic of BM is wide and lacks a clear definition. EP7-EP9 was used to understand the context for effectiveness and efficiency as related to business modeling.

Appendix D. Quotes of purpose, benefit and challenges

Table D lists the quotes of purposes, benefits, and challenges for business models and business modeling, extracted from the selected studies (see Appendix A for paper references). All quotes have been categorized into common areas (first column), and then listed under respective primary context they are found in. We use prefix notation (+) for benefit, (−) for challenge, and [PID] for the paper reference.

Table B. Inclusion and exclusion criteria

Criteria	Evaluate (=Yes)	Reasoning
EC1	Exclude if not written in English	Must be able to read and understand to evaluate
EC2	Exclude if not peer-reviewed	Basic quality assurance of paper
EC3	Exclude if duplicated	Snowballing will give many duplicates
IC1	Does the abstract, introduction, conclusions (or full text if needed) mention purposes, benefits or challenges (PBC) for business modeling?	Papers must identify real problems and issues related to business model, business modeling or business model innovation.
IC2	Does the text mention aspects of business flexibility (BF)?	BM is becoming increasingly complex due to growing business ecosystems and the digitalization of the value delivery, which both introduce a need for variability in the offering. Offering services on top of products are one example to address BF.
IC3	Does the text mention aspects of variability in the realization (VR)?	Planning a business model is not enough. It needs to be efficiently realized as well, so the business flexibility needs to be matched with a variability in the realization of the business model. Offering Software Product lines (SPL) or Product Service Systems (PSS) are examples of addressing VR.
IC4	Is it an empirical study?	We want to investigate how business models are used in practice, and not only in theory. Empirical is done in an industrial context, no student work, no proof of concept, no examples even if they are "based on real data"
IC5	Is it referring to a SIPD context?	The realization of business models is highly dependent on software due to the digitalization of the value delivery. This opens up new opportunities for value capture (and value creation) in the business ecosystems.

Table C. Data extraction properties

Id	Evaluate	How	RQ mapping
EP1	Research methods	Action research, case study, conceptual analysis, design science research, experiment, interview, literature review, not stated, other	relevance of paper
EP2	Paper context	SW intensive, industry, general (e.g. literature review), non-industry (in priority order)	RQ1 and relevance
EP3	Rigor & relevance of the paper	Detailed rubric definitions per aspect [96] Rigor: Context is described Rigor: study design is described Rigor: validity is discussed Each rigor aspect measurement: strong description (1), medium description (0.5), and weak description (0) Relevance: context (weight=8), i.e. in industrial setting Relevance: scale (weight=4), i.e. realistic size and industrial scale Relevance: subjects (weight=2), i.e. industry professionals Relevance: research method (weight=1) Each relevance aspect measurement: contribute to relevance (1), do not contribute to relevance (0)	Overview and relevance
EP4	The relevance of the paper content in respect to business modeling.	Coded 1-3: (1) business modeling; the paper discuss specifically the process of modeling your business (2) business model; the paper mainly focus on the business model and discuss how different aspects of the Business model constructs are developed (3) Other; it only refers to a specific business model(s), or discuss specific instances thereof, or a topic related to business model (e.g. flexibility); therefore of minimal significance to our study	RQ1
EP5	IC1-IC3	Use ATLAS TI to extract related quotes for each RQ.	RQ1, RQ2
EP6	Business element context	Use ATLAS TI to extract related quotes referring to a part of the business model construct, what it is, why it is important and how it is used and relates to other parts.	RQ1
EP7	Practice/technique	Use ATLAS TI to extract quotes referring to a practice or technique presented, described or used.	RQ1, RQ2
EP8	Measurement perspective	Use ATLAS TI to extract quotes related to – Product view (how well is the value created) – Process view (how efficient have you organized the value flow) – Resource view (how well is the resource utilized and adapted for the needed task) – Project view (how efficient is the goal fulfilment) – Relationship view (how effective is the communication)	RQ2
EP9	Success indicator and metric	Use ATLAS TI to extract related quotes	RQ2

Table D. Quotes on purpose, benefits and challenges for BM

Common areas	Strategy & planning (define)	Daily operations (execute)	Governance & communication
Value creation, value capture	Conceptual discussion and visualization of value creation/capture [P2] Articulate Value proposition [P7], [P13], [P35] Identify a market segment and value chain [P7], [P13], [P20] Appropriate value from technology [P36] (+) depicts the logic for value creation/capture [P17] (+) fosters innovation and increases readiness for future [P32] (+) rigorously describes and analyses business with system dynamics [P36] (−) hard managing tension between value creation and value capture (trade-offs monetization) [P5] (−) hard managing service flexibility (segmentation, QoS) [P5], [P24] (−) ensure consistent service experience (multi-channels) [P5] (−) a total value need consideration (not only financial) [P53]	Reconfiguration of roles and relationships [P8], [P20] Determining the logic for value [P30] (+) captures how resources transforms into customerswillingness to pay for value [P18] (−) service vs. product centric create conflicts, balancing is difficult [P1], [P24] (−) low effectiveness (customer experience) of value co-creation (organization/customer) [P5] (−) it is difficult to incorporate closer customer interaction [P24] (−) how to acquire resources in value chain not previously available in-house [P24]	Describe and classify businesses [P32], [P22] Meeting customerś needs [P58] Compare value creation approaches [P32] (+) facilitates strategic discussion and finding creative solutions [P2] (+) it is a structural template for mapping existing value logic [P17] (+) reduces imitability, create sustainable advantage [P24] (+) creates novel approach for using services in value creation [P41] (+) it is explicative and predictive power to value creation [P45] (+) helps calculate technology value to investors, customers, partners [P45] (−) complex coordination for ecosystem collaboration [P2] (−) negatively influences optimal value co-creation in aligned processes [P5] (−) new value (co-)creation focus on relationship-centric aspects [P7] (−) difficulty in identifying market opportunities due to changing customer needs [P9] (−) difficulty to effectively communicate (articulate, visualize) emerging value proposition [P24] (−) hard to analyse business process vs. value activities [P35] (−) many frameworks has many deficits concerning consistency and value activities [P35] (−) lacks a quantitative way to convey value and no sales model for perceived value [P48] (−) difficult to visualize value for integrated offers [P48] (−) BM has a dual nature conceptualizing value and organizing for that value (in different life cycles) [P51]

Common areas	Strategy & planning (define)	Daily operations (execute)	Governance & communication
Cost, revenue, profit	Estimate cost/revenue potential [P7] (+) depicts actual structures for a company to profit from business [P9] (+) experiment with cost before investing [P18] (−) "black-hole" investment [P18] (−) incorporate requirements for lean consumption and achieve the objectives of service profit chain [P5] (−) develop technology innovations in an adaptive process (trial-and-error) with cost as main cause for readjustments [P51]	(−) adaptation to environment by trial-and-error [P51] (−) amount of human resources needed for modeling [P56] (−) new revenue streams driven primarily by customer perceived value instead of internal cost [P24]	Incentives to engage in and control operations [P20] (−) maintain accurate definition of ownership conditions in a collaborative business model, and revenue model considering risk distribution [P54] (−) maintain a new value chain reward system [P24]
Mind-set, Knowl-edge	Experimenting [P2], [P22], [P49] Shift companyś boundaries [29] Exploit business opportunity [P22], [P29] Foster Innovation [P32] Increase knowledge [P29] (+) focus beyond company-centric focus [P17] (+) shifts focus from WHAT resources to HOW to use them [P18] (+) BMI enables strategic renewal [P36] (−) turns shared meaning into identity lock-ins [P17] (−) resistance to change [P17] (−) plan for "experimentation and learning" in established companies [P18] (−) systematic servitization (product to service shift) [P24] (−) hard to define business requirements (lack of information and specific details) [P56]	Enhance creativity, unlock barriers of innovation [P2] Build trust [P2] Increase readiness via portfolios and simulation [P9], [P32] Build knowledge [P22] (+) uses of mixed techniques between Business and IT improved communication and IT development [P56] (−) how to achieve organizational and customer learningś incorporated into iterative design [P5]	Mediating, facilitating and sharing strategic discourse [P17], [P36] Address lack of knowledge [P45], [P22] (+) unlocks barriers of innovation + building trust [P2] (+) breaks cognitive structures and act as communicative, mediating device for shared meaning and commitments [P17], [P32] (+) improves understanding, language and legitimacy [P17], [P32] (+) formalization forces implicit understanding becoming explicit (move strategy into execution) [P17] (−) lack of formality and analyst dependency with high skills [P56] (+) promotes outside in view on customer value [P18] (+) provides early warning for threatened BM via analysing dynamism of completive advantage [P18] (+) highlights consistency strategy and BM building blocks [P24] (+) provides new insights (externalize, map and store knowledge) [P32] (+) fosters systematic BMI [P32] (+) unambiguously defines dimensions, properties and semantics [P33] (+) visualization improves understanding [P32], [P56] (+) helps define goals [P32] (+) educates decision-makers for informed decisions, goals and requirement engineering [P32]

Common areas	Strategy & planning (define)	Daily operations (execute)	Governance & communication
Means	Innovation and technology management [P29] Plan and design business logic [P32] Understand complex interplay [P31] Adopt servitization to further enhance global competiveness [P54] (+) Prepares implementation (identifying joint activities with priority and validating the business model) [P2] (+) Helps to build better strategies (e-business) [P32] (−) Business model design requires better integration with strategy analysis [P37] (−) Difficult to be systematic (too slow, too detailed, iterative) [P17] (−) limited empirical validation [P17] (−) provides good insights but lacks support where to start investing to reach future business [P18] (−) capture customerś reaction to new technology [P5] (−) hard to effectively balancing (conflicting) requirements (user and design) and strategic interests (of partners) [P39] (−) tools conceptual, complicated and too time consuming (for network centric BM) [P53] (−) paradigm shift business activities and consumption patterns must be aligned with environmental and social objectives [P53]	Change and implement business logic (and business process execution) [P17], [P32] Realize strategic tasks [P9] Support resource fluidity [P15] Commercialize ideas & technology [P29] (+) better requirement engineering [P32] (+) facilitates and improves choices in IS/IT [P32] (−) difficult to mobilize and align available resources (not only internal but also extending external base) in time [P9], [P15], [P24] (−) integration, agility and change [P10] (−) barriers to change business model are real processes and tools are not good enough [P13] (−) a structured service development process connected to the business model [P24]	Alignment of strategy, business organization and technology [P32] Manage flexibility and increase change capability [P58] (+) improves measuring, observing and comparing business logic [P32] (+) improves design of sustainable business models [P32] (+) improves alignment of strategy, organization and technology and integration business IS/IT domains [P32] (+) BM may enable strategy execution and how operational choices affect companyś performance [P37] (+) helps to react to environment change due to strategic flexibility and dynamic capabilities [P52] (−) hard to reach and maintain alignment of business model and information system model [P59] (−) value co-creation is a hard cooperative process (speed, coordination, compromise) [P8] (−) how to industrialize large-scale service offerings [P24] (−) how to avoid isolated change (relationships, value, dynamic portfolio) [P24] (−) hard to visualize, document and share basic elements due to relationships and speed of change [P26], [P32] (−) hard to achieve consistency between BM and BPM and achieve real improvements with BPM [P35] (−) lack of appropriate methods and tooling for BM integrated with BPM [P35] (−) BM design requires better integration with strategy analysis models [P37] (−) discovery of goals and rules no common process for elicitation [P56]
Ends	Describe position of company in value network [P7], [P13], [P29] Formulate competitive strategy with goals and objectives [P19], [P37] Act as receipt for the business [P22]	Operationalize strategy [P36], [P37]	Alignment of strategy, business organization and technology [P32] Act as a scale model and role model for characterization of similarities and definition of difference [P22] (+) facilitates and improves choices in IS role and structure [P32]

Common areas	Strategy & planning (define)	Daily operations (execute)	Governance & communication
Assessment	Deal with uncertainty [P2], [P52], [P54] Holistic picture of future state [P2], [P32] Explain strategic issues (value creation, competitive advantage, company performance etc.) [P36], [P29] Support Leadership unity [P15] Explore and design promising business concepts/ideas [P32], [P36], [P41] Strategy and business model innovation [P17], [P36], [P52], [P53] (+) facilitates strategic discussion with shared insights to barriers/drivers (visual + levels of details) [P2] (+) facilitates interaction to create strategic options and share mediate strategic discourse [P17] (+) help to better understand the business and its important parts [P24] (+) helps to improve planning, change and implementation (with knowledge and facilitate choice of indicators) [P32] (−) difficult managing dynamics (agility, adaptability, planning, decision) for alignment to environment and other organizations [P2], [P5], [P7], [P9], [P36] (−) different methods or patterns not aligned, no guidance how to obtain final design [P49] (−) neglects the relevance for environment – focus on model-internal consistency [P49]	Alignment of control and value parameters [P6] Mapping of business roles or interactions onto technical modules, interfaces, etc. [P6] Analyse functioning of an organization [P32] Describe use of information technology [P32] Improve the Business-IS/IT dialogue [P32], [P56] (+) managing a business model portfolio can lead to flexibility in re-organizing resources [P9] (+) low-risk experiments via simulation [P32] (−) balancing act between customer, revenue, cost, functionality (e.g. local adaptation vs. sw platform) [P1] (−) mutual alignment between steps/organizations/customers when performed iteratively and holistically [P5] (−) how to match consequences of environmental changes onto company with best fit [P9] (−) a continuously learning business model experimentation [P13] (−) business model change (hard decision, risky organizational adjustments, and collective commitment) [P15] (−) efficient management of information (explore vs. create collective understanding) is difficult [P45]	Force decisions [P2] Analyse Business model fit [P49] Bridge static view for change and performance over time [P14] Computerize DDS for better design, critique and simulation of new BMs [P32] Understand how technology is converted into market outcome [P29], [P31] Provide contextual information [P35] Identification of critical success factors and investigate performance [P41] Proof, persuasion, comparison and benchmarking [P45], [P55] (+) creates common language, shared priority and forces decisions [P2] (+) improves dealing with uncertainty (reduction by sharing, turn into advantage, enhance understanding of barriers) [P2] (−) difficult to deal with uncertainty, complexity and dynamism [P54] (+) facilitates brainstorming (today and future) and integrative (no theory bias) [P17] (+) helps reducing complexity (visual) [P32] (+) improves mutual understanding Business and IT domains [P32] (+) facilities identification of key indicators to follow execution of plan [P32] (−) difficulty in reliable monitoring of key indicators [P54] (+) BM as “scale model” demonstrates feasibility and worth to partners [P45] (−) achieve joint strategy when decisions create cross-functional/divisional conflicts [P5] (−) align social, organization, and technology (due to richness and change of knowledge economy) [P7] (−) difficult to choose from massive results regarding BM design experimentation [P18] (−) hard to identify threats to BM in time [P18] (−) managed different abstraction levels and get the details right in execution [P19], [P21] (−) requires decision-making on multiple parameters of activity systems [P21] (−) BM has a dual nature (instance vs. classification) [P22] (−) hard to overcome resistance to and awareness of need to change [P52] (−) over-estimate/false impression of your ability to change [P52]

Special Section: WASA 2017 – Workshop on Automotive Software and Systems Architectures

With the advent of software and electronics, automotive companies are enabling innovation to improve safety, security, driver experience, and driving automation. Moreover, the complexity and size of software keep growing because of future innovations, such as adaptive cruise control, lane keeping, self-learning algorithms, which all leads to autonomous driving.

Consequently, increasing the use of software over the years, introduced a paradigm shift by requiring automotive companies to develop their systems and their architecture using model-based techniques. Although model-based techniques using e.g. MATLAB/Simulink and Stateflow are being accepted in the automotive industry as standard languages and tooling, advanced and dedicated techniques for system and software architecture design are still far from being widely accepted. This, however excludes the AUTOSAR standard which defines the language for designing and configuring automotive software architectures and identifies the major architectural components of automotive systems.

The Workshop on Automotive Software and Systems Architectures, WASA, was held for the third time in co-location with the International Conference on Software Architectures, in Gothenburg, Sweden. The workshop was focused on the discussions of current trends in automotive software engineering in general, and software architectures in particular. Workshop participants discussed the advances in safety systems, autonomous driving and continuous software development.

This special section presents the best paper from the workshop, by Tarun Gupta, Erik J. Luit, Martijn M.H.P. van den Heuvel and Reinder J. Bril, titled "Experience Report: Towards Extending an OSEK-Compliant RTOS with Mixed Criticality Support". The paper addresses the problem of the growing complexity and distribution of software in modern cars. The authors provide a description of how to extend an operating system to support mixed criticality on single multi-core processors; this support reduces the need for separate processors. We hope that this paper, selected from the workshop, provides an interesting contribution for the readers interested in modern automotive software.

Miroslaw Staron Chalmers/University of Gothenburg, Sweden
Yanja Dajsuren Eindhoven University of Technology, Netherlands
Harald Altinger Audi GmbH, Germany
Yaping Luo Altran Netherlands B.V, Netherlands
Andreas Vogelsang Technische Universität Berlin, Germany

e-Informatica Software Engineering Journal, Volume 12, Issue 1, 2018, pages: 305–320, DOI 10.5277/e-Inf180112

Experience Report: Towards Extending an OSEK-Compliant RTOS with Mixed Criticality Support

Tarun Gupta*, Erik J. Luit**, Martijn M.H.P. van den Heuvel**, Reinder J. Bril**
*Sioux Embedded Systems B.V.
**Mathematics and Computer Science, Technische Universiteit Eindhoven (TU/e)
tarun.gupta@sioux.eu, e.j.luit@tue.nl, m.m.h.p.v.d.heuvel@tue.nl, r.j.bril@tue.nl

Abstract

Background: With an increase of the number of features in a vehicle, the computational requirements also increase, and vehicles may contain up to 100 Electronic Control Units (ECUs) to accommodate these requirements. For cost-effectiveness reasons, amongst others, it is considered desirable to limit the growth of, or preferably reduce, the number of ECUs. To that end, mixed criticality is a promising approach that received a lot of attention in the literature, primarily from a theoretical perspective.
Aim: In this paper, we address mixed criticality from a practical perspective. Our prime goal is to extend an OSEK-compliant real-time operating system (RTOS) with mixed criticality support, enabling such support in the automotive domain. In addition, we aim at a system (*i*) supporting more than two criticality levels; (*ii*) with minimal overhead upon an increase of the so-called criticality level indicator of the system; (*iii*) requiring no changes to an underlying operating system; and (*iv*) featuring further extensions, such as hierarchical scheduling and multi-core.
Method: We used the so-called adaptive mixed criticality (AMC) scheme as a starting point for mixed criticality. We extended that scheme from two to more than two criticality levels (satisfying (*i*)) and complemented it with specified behavior for criticality level changes. We baptized our extended scheme AMC*. Rather than selecting a specific OSEK-compliant RTOS, we selected ExSched, an operating system independent external CPU scheduler framework for real-time systems, which requires no modifications to the original operating system source code (satisfying (*iii*)) and features further extensions (satisfying (*iv*)).
Results: Although we managed to build a functional prototype of our system, our experience with ExSched made us decide to rebuild the system with a specific OSEK-compliant RTOS, being μC/OS-II. We also briefly report upon our experience with AMC* and suggest directions for improvements.
Conclusions: Compared to extending ExSched with AMC*, extending μC/OS-II turned out to be straightforward. Although we now have a basic system operational and available for experimentation, enhancements of the AMC*-scheme are considered desirable before exploitation in a vehicle.

Keywords: OSEK, RTOS, mixed criticality

1. Introduction

A growing trend in the automotive domain is a feature intensive vehicle. These features may be safety related, driver assistance related, connected services, or multimedia and entertainment related. With an increase of the number of features, the computational requirements also increase. Nowadays, a vehicle may be controlled by over 100 million lines of code, that are executed on up to 100 Electronic Control Units (ECUs) [1]. For reasons of cost, space, weight, and power con-

sumption, amongst others, adding more ECUs is undesirable. Instead, even a reduction of ECUs is preferred, with appropriate means for (temporal and spatial) isolation between applications, efficient and effective resource management, assurance against failure, and graceful degradation upon overloads. Given the distinct criticality levels of these features, e.g. safety critical, mission critical, and low-critical, application of mixed criticality theory and practice [2] may be beneficial. Within the context of the i-GAME [3] and EMC^2 projects[1], we therefore explored the option to apply mixed criticality. Whereas there exists an overwhelming number of papers on mixed criticality, the majority addresses theoretical aspects, with a focus on schedulability analysis. Although some address implementation aspects, such as [4], only a few present actual implementations extending an operating system, such as [5–7]. None of these aim at the automotive domain, however, which is the main focus of this paper.

In this paper, we report upon our initial efforts to extend an OSEK-compliant [8] real-time operating system (RTOS) for a single-core with support for mixed criticality. In addition, we aim at a system (*i*) supporting more that two criticality levels; (*ii*) with minimal overhead upon an increase of the so-called criticality level indicator of the system; (*iii*) requiring no changes to an underlying operating system; and (*iv*) featuring further extensions, such as hierarchical scheduling and multi-core.

For our mixed criticality scheme, we selected an existing scheme, Adaptive Mixed Criticality (AMC) [9], as a basis. We extended the scheme from two criticality levels to multiple criticality levels (satisfying (*i*)), and complemented it with specified behavior upon criticality level changes. Rather than selecting a specific OSEK-compliant operating system, we selected ExSched [10]. ExSched is an operating system independent external CPU scheduler framework for real-time systems, which requires no patches (i.e. modifications) to the original operating system source code (satisfying (*iii*)), unlike, for example LITMUS$^{\mathrm{RT}}$ [11] and AQUOSA [12], making it easier to update to newer kernel versions. Moreover, ExSched supports multiple operating systems, in particular Linux and VxWorks, and comes with hierarchical and multi-core schedulers (satisfying (*iv*)), amongst others. In our initial experiments, we used ExSched in combination with Linux version 2.6.36, which we downloaded from [13], on an Intel Core I5 processor. We intended to subsequently develop support of ExSched to support an OSEK-compliant RTOS. Although we managed to build a functional prototype of our system, we decided to abandon ExSched, however, based on our experiences with and insights gained during the extension of ExSched with mixed criticality support. Our subsequent experiments therefore concerned the move from ExSched with Linux towards an OSEK-compliant RTOS, in particular μC/OS-II [14]. We used μC/OS-II in combination with RELTEQ (Relative Timed Event Queues) [1], which supports hierarchical scheduling (*iv*), amongst others. Our final contribution concerns a reflection on AMC*.

This journal paper is an extended version of a workshop paper [15]. Compared to [15], this extended version has the following two major contributions. Firstly, it presents the extension of μC/OS-II with mixed criticality (Section 6.2). Secondly, it presents the experience with and the evaluation of AMC*, including improvements of the scheme (Section 7).

The remainder of this paper is organized as follows. We start by a brief discussion of related work in Section 2. Next, in Section 3, we present our real-time scheduling model and a brief recapitulation of ExSched. Our extended AMC scheme, baptized AMC*, is the topic of Section 4. Extending ExSched with mixed criticality support is addressed in Section 5. The move from ExSched with Linux to μC/OS-II is addressed in Section 6. In Section 7, we reflect on AMC*. The paper is concluded in Section 8.

[1]The work presented in this paper was funded in part by the EU 7^{th} framework programme through the i-GAME (Interoperable GCDC AutoMation Experience) project (grant agreement 612035) and the ARTEMIS Joint Undertaking EMC^2 project (grant agreement 621429).

2. Related work

There exists a plethora of papers on mixed criticality systems; see [2] for a review. Here, we focus on two specific mixed criticality aspects, being mixed criticality schemes and actual implementations extending an operating system, and briefly discuss existing support of OSEK-compliant RTOSs.

Building upon the seminal work of Vestal [16], which was the first paper addressing schedulability analysis for mixed criticality systems given a basic mixed criticality scheme, a lot of theoretical work has been done on mixed criticality systems. The scheme presented by Vestal consists of an ordered set of four criticality levels. At any moment of time, the system is running at a particular criticality level. The scheme describes the cause (i.e triggering event) and behavior of a criticality level up, i.e. when the system makes the transition from a lower to a higher criticality level, but lacks a description of a criticality level down. This initial scheme was later refined and the schedulability of mixed criticality systems improved by Baruah et al. in [4, 9, 17], amongst others. Although the restriction on the number of criticality levels is lifted in these later works, the description of the latest scheme [9] called adaptive mixed criticality (AMC) and its analysis has been restricted to two levels for simplicity. The cause and behavior of a criticality level down was first described in [4]. The AMC scheme was later relaxed in [18] at the cost of increased implementation complexity. In this paper, we therefore selected AMC as a basis. For a detailed comparison of the schemes mentioned above, the interested reader is referred to [19, 20].

Whereas a lot of papers address theoretical aspects, only a few papers describe actual implementations extending an operating system with mixed criticality support, such as [5–7]. Kim et al. [6] studied the actual implementation of a criticality level change in the RTOS eCOS [21], with the aim to minimize the scheduler overheads. They assume a mixed criticality scheme with two criticality levels. Herman et al. [5] describe RTOS support for multi-core mixed criticality systems, using the academic RTOS LITMUS$^{\mathrm{RT}}$ [11], an extension to the Linux kernel. The number of criticality levels assumed in that work is four. Kritikakou et al. [7] describe support for multi-core mixed criticality systems using their own developed bare-metal library [22]. In their model, only a single task of a high criticality level is assumed. To the best of our knowledge, there does not exist an OSEK-compliant [8] RTOS with an extension for mixed criticality, which is the focus of this paper.

There exist many OSEK-compliant RTOSs, such as ETAS RTA-OSEK[2], μC/OS-II [14], and Erika Enterprise RTOS[3]. The specification of the OSEK operating system [8] explicitly states that the "*OSEK operating system is a single processor operating system meant for distributed embedded control units*". An OSEK-compliant RTOS may therefore provide support for hierarchical scheduling and multi-core, but need not provide such support. As examples, both ETAS RTA-OSEK and μC/OS-II provide neither hierarchical scheduling nor multi-core support, whereas Erika Enterprise RTOS only provides multi-core support. An extension of μC/OS-II with hierarchical scheduling has been described in [23]. To the best of our knowledge, there does not exist an OSEK-compliant RTOS providing support for both hierarchical scheduling and multi-core. In this paper, we focus on μC/OS-II, because we gained significant experience with that RTOS over the past years [24–26].

3. Preliminaries

In this section, we present our real-time scheduling model in Subsection 3.1 and a brief recap of ExSched in Subsection 3.2.

3.1. Real-time scheduling model

After presenting a basic real-time scheduling model for fixed-priority pre-emptive scheduling (FPPS), we extend the model with mixed criticality conform AMC [9].

[2]Details about ETAS RTA-OSEK can be found at http://www.etas.com.
[3]Details about Erika Enterprise OS can be found at http://www.tuxfamily.org.

3.1.1. Basic model for FPPS

We assume a single processor and a set $\mathcal{T}$ of n independent sporadic tasks τ_1, τ_2, ..., τ_n, with unique priorities π_1, π_2, ..., π_n. At any moment in time, the processor is used to execute the highest priority task that has work pending. For notational convenience, we assume that (i) tasks are given in order of decreasing priorities, i.e. τ_1 has the highest and τ_n the lowest priority, and (ii) a higher priority is represented by a higher value, i.e. $\pi_1 > \pi_2 > \cdots > \pi_n$.

Each task τ_i is characterized by a *minimum inter-activation time* $T_i \in \mathbb{R}^+$, a *worst-case computation time* $C_i \in \mathbb{R}^+$, and a (*relative*) *deadline* $D_i \in \mathbb{R}^+$. We assume that the constant pre-emption costs, such as context switches, are subsumed into the worst-case computation times. We assume constrained deadlines, i.e. the deadline D_i may be smaller than or equal to period T_i. The *utilization* U_i of task τ_i is given by C_i/T_i, and the *utilization* U of the set of tasks $\mathcal{T}$ by $\sum_{1\leq i\leq n} U_i$.

We also adopt standard basic assumptions [27], i.e. tasks do not suspend themselves and a job does not start before its previous job is completed.

3.1.2. Extended model for mixed criticality

We assume a set $\mathcal{L}$ of m criticality levels[4] Λ_1, Λ_2, ..., Λ_m. For notational convenience, we assume that (i) criticality levels are given in order of decreasing criticality, i.e. Λ_1 represents highest and Λ_m represents lowest criticality, and (ii) a higher criticality level is represented by a higher value, i.e. $\Lambda_1 > \Lambda_2 > \ldots > \Lambda_m$.

Each task τ_i has a particular criticality level $\lambda_i \in \mathcal{L}$, termed its *representative* criticality level. We now define subsets $\mathcal{T}^\Lambda$ of $\mathcal{T}$, i.e.

$$\mathcal{T}^\Lambda \stackrel{\text{def}}{=} \{\tau_i | \lambda_i \geq \Lambda\} \tag{1}$$

When the system is executing at criticality level Λ, i.e. the criticality level indicator Γ is equal to Λ, the processor is used to execute only tasks in the subset $\mathcal{T}^\Lambda$.

Moreover, the worst-case computation time of a task τ_i becomes a vector $\vec{C}_i$ indexed by criticality level. These computation times are monotonically non-decreasing for increasing criticality levels, i.e.

$$\Lambda_k \leq \Lambda_\ell \leq \lambda_i \Rightarrow \vec{C}_i(\Lambda_k) \leq \vec{C}_i(\Lambda_\ell) \tag{2}$$

The actual execution time of the current job of task τ_i at time t is denoted by $\beta_i(t)$.

The following condition defines when a criticality level up occurs[5].

Condition 1. *When a job of task τ_i is executing at time t while the system is running at level Λ with $\Lambda \leq \lambda_i < \Lambda_1$ and the actual execution time $\beta_i(t)$ equals the worst-case computation time $\vec{C}_i(\Lambda)$ of τ_i, a criticality level up occurs.*

A criticality level down occurs upon a so-called criticality level Λ idle time.

Definition 1. *A criticality level Λ idle time is an instant at which there is no pending load of the tasks in $\mathcal{T}^\Lambda$.*

Intuitively, a task has *pending load* [28] larger than zero at time t when it has been activated strictly before time t and did not complete yet at time t.

Condition 2. *Upon a criticality level Λ idle time with $\Lambda > \Lambda_m$ a level down change occurs.*

3.2. Recapitulation of ExSched

The ExSched framework [10] is a loadable Linux kernel module and an extension of the REal-time SCHeduler (RESCH) framework [29]. Figure 1 shows the structural components of ExSched. An application uses the ExSched APIs provided by the ExSched Library in user space to communicate with the main ExSched Module in kernel space. This communication takes place via the ioctl () system call, i.e. ExSched is built as a character-device module. The ExSched framework supports development of plug-ins with the help of callback functions. Plug-ins for hierarchi-

[4] In [9], a so-called *dual-criticality system* is assumed, i.e. $m = 2$. In this paper, we assume more than 2 criticality levels.

[5] The AMC scheme assumes that a criticality level up is handled instantaneously.

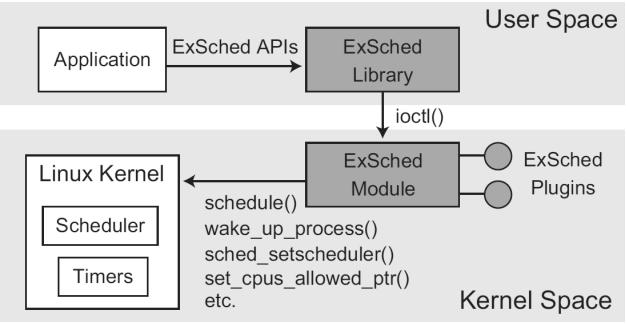


Figure 1. ExSched: structural components [10]

cal scheduling and multi-core scheduling are part of ExSched's release.

The ExSched Library provides methods to

- register/de-register tasks: rt_init() and rt_exit();
- set parameters of tasks: rt_set_wcet(), rt_set_period(), rt_set_deadline(), and rt_set_- priority();
- start a task: rt_run(); and
- activate a next job, i.e. wait (sleep) until the next period: rt_wait_for_period().

The ExSched Module uses the POSIX-compliant SCHED_FIFO scheduling policy provided by the Linux kernel. The module maintains its own task structure, which extends the encapsulated Linux task structure with additional timing parameters provided through the ExSched Library. The ExSched Module provides a dedicated interface to install and un-install a plug-in; see Table 1. Only one plug-in can be installed in ExSched at the time.

4. AMC* scheme

In Subsection 3.1.2, a general model for mixed criticality has been given, leaving specific details unspecified, such as (*i*) what happens when a task τ_i exceeds its worst-case computation time at its representative criticality level, (*ii*) what will be the new criticality level at which the system will execute upon a criticality level change, and (*iii*) what happens with the (jobs of the) tasks that are no longer executed when a criticality level up occurs and accordingly how to deal with tasks that are again allowed to execute when a criticality level down occurs. In this section, we consider these three topics for our AMC*-scheme.

4.1. Overrun of $\vec{C}_i(\lambda_i)$

For AMC*, we consider an overrun of the worst-case computation time of a task τ_i at its representative criticality level λ_i *erroneous behavior*, similar to [18]. Upon such an overrun, a criticality level up occurs if $\lambda_i < \Lambda_1$. The behavior is unspecified for $\lambda_i = \Lambda_1$; see also Condition 1.

4.2. New criticality level upon a criticality level change

Because an overrun at criticality level Λ_1 is considered erroneous behavior and worst-case computation times are monotonically non-decreasing for increasing criticality levels (2), we consider three cases when a criticality level up occurs, assuming the system is executing at criticality level Λ:

1. $\Lambda \leq \lambda_i < \Lambda_1 \wedge \vec{C}_i(\Lambda) = \vec{C}_i(\lambda_i)$: When a task τ_i overruns its worst-case computation time at its representative criticality level λ_i and λ_i is smaller than the highest criticality level Λ_1, the new criticality level Λ^{new} is the smallest criticality level larger than λ_i, i.e.
$$\Lambda^{\text{new}} = \min\{\lambda \in \mathcal{L} | \lambda > \lambda_i\} \qquad (3)$$
2. $\vec{C}_i(\Lambda) < \vec{C}_i(\lambda_i)$: When a task τ_i overruns its worst-case computation time at the criticality level Λ ($\vec{C}_i(\Lambda)$) and that computation time is less than its worst-case computation time at its representative criticality level λ_i ($\vec{C}_i(\lambda_i)$), the new criticality level Λ^{new} is the smallest

Table 1. Existing API provided by the ExSched Module to install and un-install a plug-in

Method	Description
extern void install_scheduler(void (*task_run_plugin)(resch_task_t*), void (*task_exit_plugin)(resch_task_t*), void (*job_release_plugin)(resch_task_t*), void (*job_complete_plugin)(resch_task_t*));	Install Plug-in
extern void uninstall_scheduler(void);	Un-install Plug-in

criticality level not giving rise to an overrun for τ_i, i.e.

$$\Lambda^{\text{new}} = \min\left\{\lambda \in \mathcal{L} | \vec{C}_i(\Lambda) < \vec{C}_i(\lambda)\right\} \quad (4)$$

3. **Unspecified behavior** An overrun of $\vec{C}_i(\lambda_i)$ of task τ_i is unspecified for $\lambda_i = \Lambda_1$; see Subsection 4.1.

When a criticality level down occurs, the system returns to the lowest criticality level Λ_m.

4.3. Policies for criticality level changes

We considered three policies for mixed criticality, i.e. *suspend*, *resume*, and *abort*.

Definition 2. *The* suspend *policy for a task (*i*) temporarily does not give any execution time to a currently active job of that task and (*ii*) suppresses new releases of jobs of that task.*

Definition 3. *The* resume *policy allows suspended tasks to release new jobs.*

The release of new jobs shall satisfy the constraints of the system, i.e. no earlier than allowed according to the minimal inter-activation time.

Definition 4. *The* abort *policy for a task decides whether or not the current job of a suspended task is discarded or allowed to continue at a later time.*

The *abort* policy is conditional, i.e. depending on the context, suspended jobs of a task may, but need not, be aborted. As an example, in a reservation-based resource management context, where suspension is used to prevent jobs of tasks to execute upon depletion of a budget, abortion will not be applied. In our initial experiments extending ExSched with mixed criticality, we suspend jobs of tasks that are no longer executed when a criticality level up occurs and abort those jobs when a criticality level down subsequently occurs. By delaying the actual abort, we minimize overhead upon a criticality level up.

5. Extending ExSched with mixed criticality support

In this section, we describe our extension of ExSched with mixed criticality support. We start with a description of the required extensions in Section 5.1. The design of the system is the topic of Section 5.2. We demonstrate our implemented system by means of an example in Section 5.3.

5.1. Basic mechanisms

To support the AMC* scheme, the following basic mechanisms, are required:

- *run-time monitoring*, to keep track of the amount of time a job of a task has spent on execution, to detect depletion of a "*budget*", and to realize (i.e. trigger the handler for) the criticality level up functionality;
- *task-management services*, i.e. the *suspend*, *resume*, and *abort* policies, which have been described in Subsection 4.3;
- *idle-time detection*, to realize (i.e. trigger the handler for) the criticality level down functionality.

We briefly consider these mechanisms in the following subsections.

5.1.1. Run-time monitoring

Run-time monitoring is a basic mechanism that is not only required for mixed criticality, but also for reservation-based resource management.

Rather than incorporating run-time monitoring in a to-be-developed AMC∗ plug-in, we therefore decided to extend the ExSched Module.

The timers used in ExSched do not satisfy our needs, however. In particular, the values of the execution times are stored in so-called "*jiffies*", which is the time between two successive clock ticks of the real-time clock. Instead of using the (low-resolution) real-time clock, we decided to base monitoring on high-resolution timers (provided through hrtimer.h), with a resolution in the order of nanoseconds on an Intel Core I5 processor. The methods for run-time monitoring are described in Table 2.

5.1.2. Task management services

The task management functionality to realize criticality level changes is described in Table 3. We believe that this functionality is of a generic nature, i.e. that it can also be used by other plug-ins. Moreover, in order to be able to "*hide*" the specific details of the actual operating system, which is one of the design goals of ExSched [10], plug-ins shall not be aware of specific operating system functionality. As a result, the ExSched Module is the only place where this functionality can be implemented.

Note that we combined the *resume* and *abort* policy into a single primitive resume_task(). For the AMC* implementation described in this paper, we always pass a value true for the parameter abort when calling resume_task().

The abort functionality has been implemented using Linux signals, in particular the POSIX compliant SIGUSR1 signal. Before a task is actually resumed and only when a job of the task has been suspended, a SIGUSR1 is sent to the task. This allows the task to perform clean-up activities as required when resumed.

5.1.3. Extended plug-in interface

Table 4 presents the methods that the AMC∗ plug-in provides to the ExSched Module, allowing the latter to bring criticality level up and criticality level down events to the attention of the plug-in. We expect the methods to be of a sufficient generic nature to justify incorporation in the generic install-methods, e.g. to monitor individual tasks. The description given in Table 4 is therefore from the perspective of the AMC∗ plug-in. The method to install a plug-in given in Table 1 is extended with parameters for these two methods.

5.2. System design

Figure 2 shows the static structure of ExSched extended with AMC*. A similar structure has been used for AMC* as for ExSched, i.e. a library AMC∗ Library in user space and a loadable kernel module AMC∗ Plugin Module in kernel space, using the ioctl () system call for communication. Although we generalized ExSched by extending the ExSched Module with run-time monitoring and additional task-management services, amongst others, its general architecture remained unchanged, i.e. Figure 2 is an extension of Figure 1.

The generic interfaces of the ExSched Module towards a plug-in have been described in the previous section. Below, we consider the AMC∗ Library and the AMC∗ Plugin Module in more detail. Both modules share a header file defining a constant NO_OF_CRIT_LEVELS denoting the number of criticality levels supported by the system.

5.2.1. AMC∗ Library

The AMC∗ Library provides a method to allow a task τ_i to set its representative criticality level λ_i and its worst-case computation times for each criticality level $\lambda \in \mathcal{L}$; see Table 5.

5.2.2. AMC∗ Plugin Module

The AMC∗ Plugin Module stores the representative criticality level λ of each task and the worst-case computation time $\vec{C}$ of each task for every criticality level. Moreover, it stores and maintains the criticality level indicator Γ of the system. In particular, it implements the handlers for the criticality level up and criticality level

Table 2. New local methods of the ExSched Module for run-time monitoring

Method	Description
void start_monitor_timer (resch_task_t *rt)	Start timer for a task τ_i denoted by *rt for an amount of time $\vec{C}_i(\Lambda) - \beta_i(t)$. Both $\vec{C}_i(\Lambda)$ and $\beta_i(t)$ are stored in the task's control block.
void stop_monitor_timer (resch_task_t *rt)	Stop timer for a task τ_i denoted by *rt and update $\beta_i(t)$.
enum hrtimer_restart monitor_expire_handler (struct hrtimer *timer)	Interrupt handler of the timer identified by *timer.

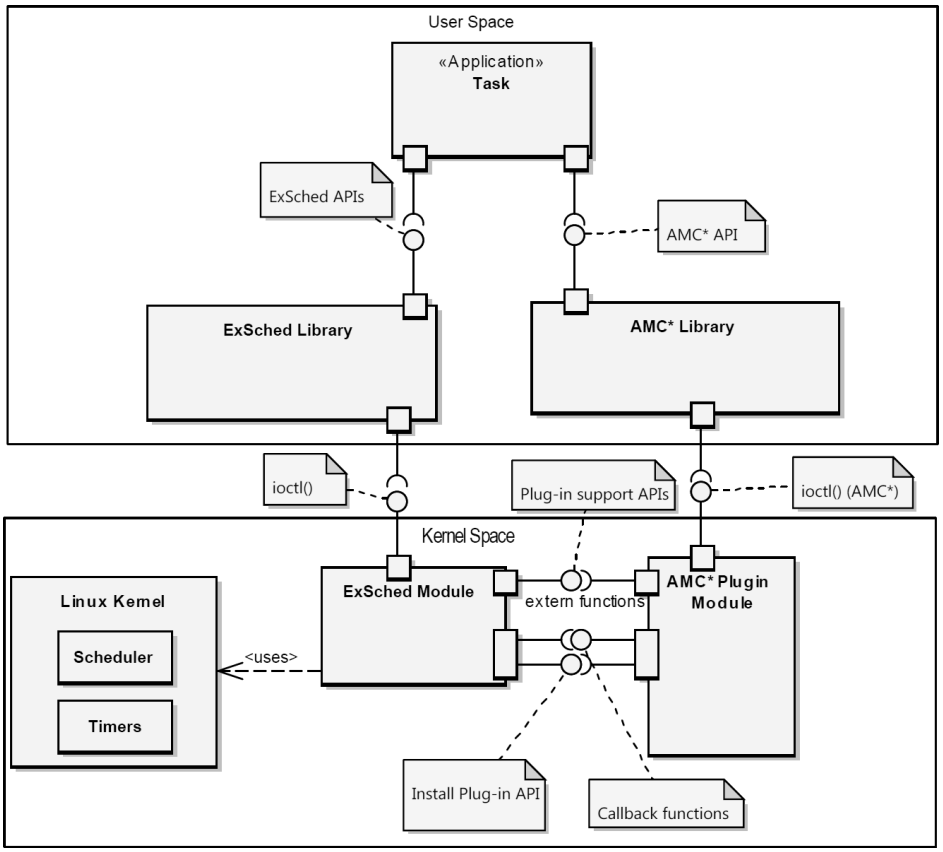


Figure 2. ExSched extended with AMC* [19, 20]

down handler, using the functionality provided by the ExSched Module.

5.3. An example

In this section, we illustrate our system by means of an example[6], with 3 criticality levels and 4 tasks. The characteristics of the synthetic task set are given in Table 6[7].

Figure 3, which has been created by means of Grasp [30][8], shows a timeline with the executions of the tasks. The figure shows both a criticality level up, at time 57 ms and 61 ms, as well as a criticality level down at time 73 ms.

[6]The interested reader is referred to [19, 20] for other examples.

[7]Don't-care values for $\vec{C}$ are specified as zero, i.e. $\lambda_i < \lambda \Rightarrow \vec{C}_i(\lambda) = 0$.

[8]A version of Grasp is available in the ExSched distribution at http://www.idt.mdh.se/~exsched/.

Table 3. New methods provided by the ExSched Module to its plug-ins for task management

Method	Description
void suspend_task (resch_task_t *rt)	Suspends a task.
void resume_task (resch_task_t *rt, bool abort)	Resumes a task. When abort is true, a pending job will be aborted.
void abort_job (resch_task_t *rt)	Aborts the pending job of a task.

Table 4. New methods expected by the ExSched Module from its plug-ins, i.e. callback functions, to handle criticality level changes

Method	Description
void (*monitor_expire_plugin) (resch_task_t *rt)	Criticality-level up handler.
void (*idle_time_plugin) (resch_task_t *rt)	Criticality-level down handler.

Table 5. AMC∗ API: Method provided by the AMC∗ Library

Method	Description
int rt_set_rep_crit_level (int rep_crit, unsigned long[NO_OF_CRIT_LEVELS] wcet_per_crit)	Method to set a task's representative criticality level and worst-case computation time per criticality level.

6. Moving from Linux to μC/OS-II

ExSched [10] supports both Linux and VxWorks, but lacks support for an OSEK-compliant real-time operating system, such as μC/OS-II or ERIKA Enterprise [31]. In this section, we describe our efforts in moving from ExSched with Linux to μC/OS-II. We start this section with our experience with and evaluation of ExSched. We subsequently consider the usage of μC/OS-II.

6.1. Experience with and evaluation of ExSched

Based on ExSched's features, i.e. (*i*) being an operating system independent external CPU scheduler framework, (*ii*) providing support for temporal isolation through hierarchical scheduling, and (*iii*) providing support for multi-core scheduling, selecting ExSched for our extension with support for mixed criticality seemed a good choice. As illustrated by the example in Section 5.3, we managed to build a functional prototype of our system.

Extending ExSched with mixed criticality support turned out to be laborious, however. Instead of adding just a "*mixed criticality*"-specific plug-in, we also extended and revised the ExSched Module, as described in Section 5. Although the code is documented with samples illustrating its usage, critical user documentation is missing. A conference paper [10] describes Exsched's high-level design. Other software engineering artifacts, such as requirements, specification, design and corresponding tests are unavailable. As a result, the Exsched API is hard to validate and testing our mixed criticality plugin against its API is even harder. Although we resolved the problems with Exsched that we encountered, we did not thoroughly validate and verify the existing functionality of ExSched. Based on our experience, the framework is hard to maintain.

Table 6. Task characteristics

Task	λ	π	$\vec{C}$ Λ_3	Λ_2	Λ_1	T
Task 1	Λ_3	99	6 ms	0 ms	0 ms	45 ms
Task 2	Λ_2	98	6 ms	10 ms	0 ms	50 ms
Task 3	Λ_2	97	6 ms	6 ms	0 ms	50 ms
Task 4	Λ_1	96	6 ms	9 ms	12 ms	60 ms

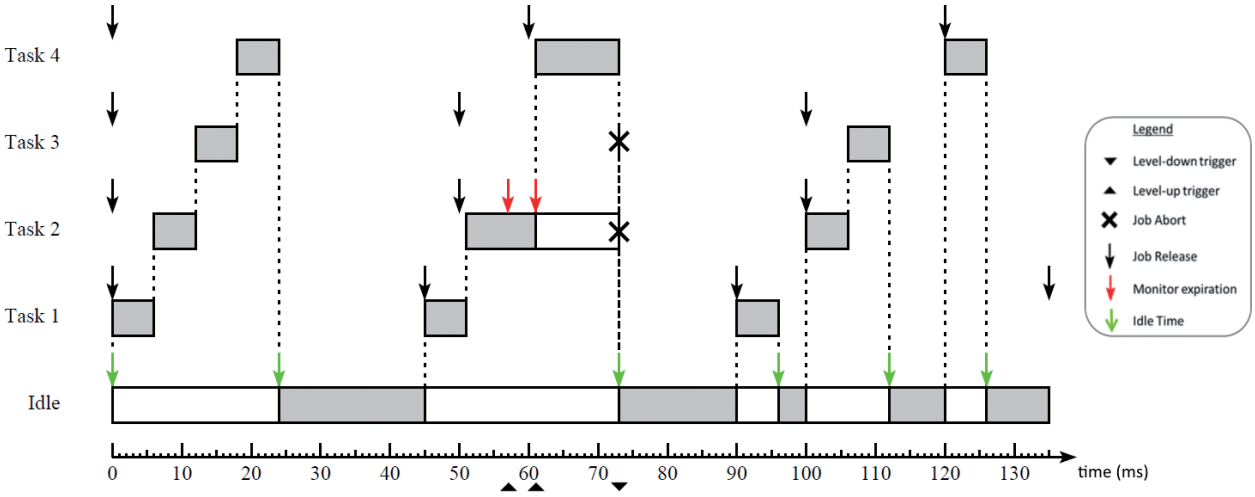


Figure 3. At time 57 ms, the job of Task 2 executed for its worst-case computation time at criticality level Λ_3 but didn't complete yet. As a result, a criticality level up change to Λ_2 occurs. At time 61 ms the job of Task 2 executed for its worst-case computation time at its representative criticality level Λ_2 but didn't complete yet, resulting in a criticality level up change to Λ_1. The active jobs of Task 2 and Task 3 at time 61 ms are suspended due to the criticality level up. A criticality level down occurs at time 73 ms to Λ_3 and the SIGUSR1 signal is sent at that time, effectively aborting the suspended job

One of the reasons to select ExSched was the availability of existing plug-ins, such as hierarchical scheduling and multi-core. As described in Section 3.2, the current implementation only allows to use a single plug-in at the time, however. Whenever multiple plug-ins are desired, a major redesign of ExSched seems to be required. Given these experiences and gained insight, we decided to entirely abandon ExSched for our future efforts.

6.2. Extending μC/OS-II with AMC*

In this section, we briefly describe the rationale for selecting μC/OS-II, the extension of μC/OS-II with AMC*, and a comparison between ExSched and μC/OS-II regarding the extension with mixed criticality.

6.2.1. Background and rationale

Based on our earlier experience with the OSEK-compliant RTOS μC/OS-II [14][9] in (*i*) research [24, 25], (*ii*) an automotive case study implementing and demonstrating active suspension in a Jaguar XF [26], and (*iii*) education, i.e. the core course *Real-time software systems engineering* (2IN70) in the master automotive technology [32] at the TU/e, we decided to select this RTOS for our next step and to use it in combination with RELTEQ (Relative Timed Event Queues) [1, 23]. RELTEQ provides a general timer management system and supports periodic tasks and a hierarchical scheduling framework (HSF) in our extended implementation of μC/OS-II; see Figure 4.

[9]Unfortunately, the supplier of μC/OS-II, Micrium, has discontinued the support for the OSEK-compatibility layer.

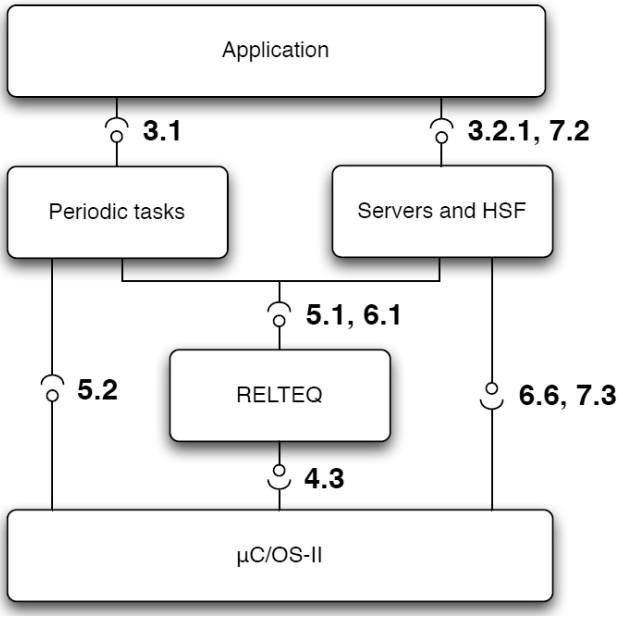


Figure 4. Interfaces between μC/OS-II and its extension [23]. The numbers indicate the sections in [23] describing the provided interfaces and their implementation

In our earlier work, we created a port for μC/OS-II to the OpenRISC platform [33] to experiment with the accompanying cycle-accurate simulator and to ease development. Our set-up also runs on a Freescale EVB9S12XF512E evaluation board with a 16-bits, MC9S12XF512 processor and 32 kB on-chip RAM.

6.2.2. Basic mechanisms and specific AMC*-functionality

As described in Section 5.1, three sets of basic mechanisms are required to support AMC*, *run-time monitoring*, *task management services*, and *idle-time detection*. All these mechanisms are essentially supported through RELTEQ and our earlier extension of μC/OS-II with an HSF.

To implement specific AMC*-functionality, we extended the task-control block with mixed-criticality specific information, such as the representative criticality level λ and the vector of worst-case computation times $\vec{C}$. In addition, we provided a method similar to rt_set_rep_crit_level (see Table 5) to set λ and $\vec{C}$. Given these mechanisms and extended data structures, implementing the specific AMC*-functionality turned out to be straightforward. In particular, we implemented a dedicated module for AMC* with the level up handler and the level down handler (see Table 4). The level up handler is called from RELTEQ, upon the detection of an overrun, and the level down handler is called from the OSTaskIdleHook within μC/OS-II. Support for run-time monitoring (Table 2) and task management (Table 3) is provided by RELTEQ and μC/OS-II, respectively. An overview of the μC/OS-II architecture including the extensions for both RELTEQ and AMC* is given in Figure 5.

6.2.3. A comparison between ExSched and μC/OS-II

Compared to ExSched with Linux, extending RELTEQ and μC/OS-II with AMC* was relatively easy. The only functionality implemented in Linux that could not be supported by our μC/OS-II extension concerned allowing a task to perform the clean-up activities as required when resumed, i.e. μC/OS-II lacks functionality similar to the POSIX compliant SIGUSR1 signals. Another disadvantage of our extension of RELTEQ and μC/OS-II with AMC* is that RELTEQ

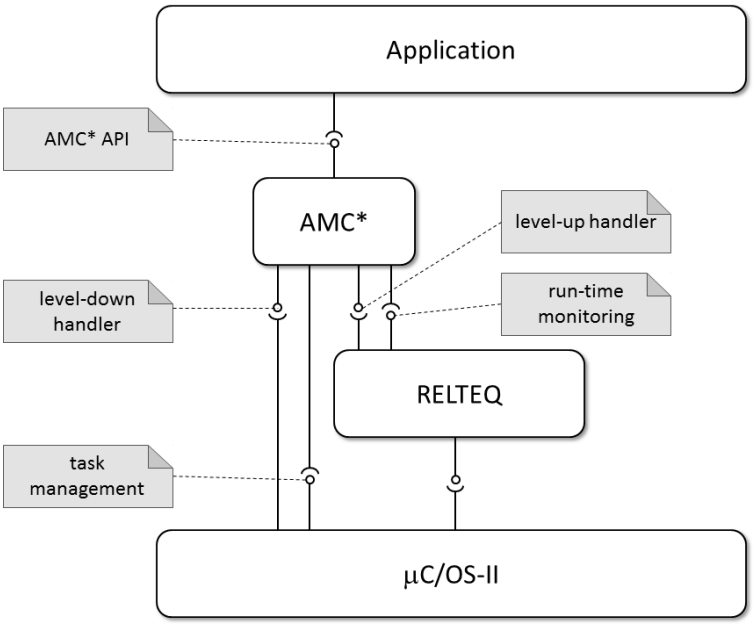


Figure 5. Interfaces between μC/OS-II, RELTEQ and AMC*

has been implemented in μC/OS-II[10], and our AMC* extension required additional changes to μC/OS-II as well. Hence, whereas ExSched requires no pathes (modifications) to the original source code of the underlying operating system, our support for AMC* using RELTEQ did require modifications to μC/OS-II. Extending μC/OS-II with AMC* without patches to the original source code would be considerably less straightforward.

As a final remark, we merely observe that whereas μC/OS-II inherently provides functionality to suspend and resume a task by means of OSTaskSuspend() and OSTaskResume(), the OSEK/VDX-standard [8] lacks such functionality. Extending an OSEK-compliant RTOS with mixed criticality without patches may therefore not be trivial.

7. A reflection on AMC*

In this section, we briefly reflect on AMC*, our mixed criticality scheme. We first report on our experience with AMC*. We subsequently describe directions for resolving the undesirable behavior encountered.

7.1. Experience with and evaluation of AMC*

Within the literature, various options for improvement of the AMC-scheme have been proposed [2, 18]. Below, we briefly report upon two aspects we encountered while experimenting with our implementation that have, to the best of our knowledge, not been reported before in the literature.

7.1.1. Erroneous and unspecified behavior

As described in Section 4.1, an overrun of the worst-case computation time of a task τ_i at its representative criticality λ_i is considered *erroneous behavior*. Moreover, the behavior is *unspecified* when $\lambda_i = \Lambda_1$. Figure 3 shows an example with erroneous behavior of a job of Task 2 with $\lambda_2 < \Lambda_1$, which gives rise to a criticality level up conform the AMC scheme (see Condition 1). A drawback of this behavior is that Task 3, which has the same representative criticality level as Task 2, is also no longer allowed to execute, and its activation at time 50 is therefore aborted as well. Moreover, this criticality level up is not

[10]Unlike the implementation of an HSF in μC/OS-II, the implementation of an HSF in VxWorks described in [34] required no changes to the operating system.

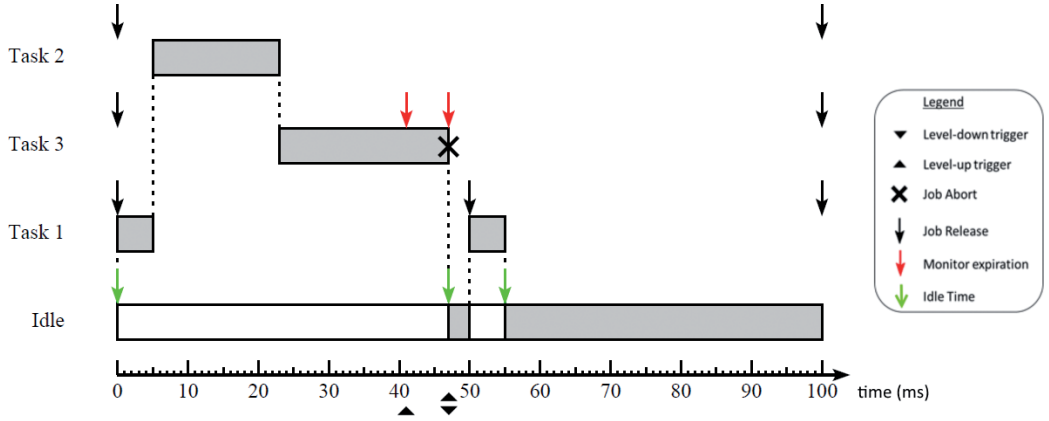


Figure 6. A criticality level up immediately followed by a criticality level down in AMC*

Table 7. Task characteristics

Task	λ	π	$\vec{C}$ Λ_3	Λ_2	Λ_1	T
Task 1	Λ_3	99	5 ms	0 ms	0 ms	50 ms
Task 2	Λ_1	98	18 ms	24 ms	24 ms	100 ms
Task 3	Λ_2	97	18 ms	24 ms	0 ms	100 ms

necessitated by a need for more anticipated resources by tasks with a higher representative criticality level than the criticality level at which the system is executing, but instead to prevent the erroneous behavior of Task 2 from jeopardizing the correct timing behavior of tasks with the same or a higher representative criticality level.

Although it is theoretically (i.e. from an academic perspective) convenient to classify an overrun of the worst-case response time of a task at its representative criticality level as erroneous behavior, this is clearly not desirable from a practical (i.e. an industrial) perspective.

7.1.2. Criticality-level up immediately followed by a criticality level down

Using the original AMC*-scheme, a criticality level up can be immediately followed by a criticality- level down, as illustrated in Figure 6 for a task set with characteristics as given in Table 7.

At time $t = 41$, a job of Task 3 experiences an overrun of $\vec{C}_3(\Lambda_3) = 18$ ms and a criticality level up occurs from criticality level Λ_3 to Λ_2. The job of Task 3 experiences a next overrun of $\vec{C}_3(\Lambda_2) = 24$ ms at time $t = 47$ ms and a criticality level up occurs towards Λ_1. The system subsequently encounters an idle-time and the system returns to its lowest criticality level Λ_3, i.e. the system exhibits the undesirable behavior of a criticality level up immediately followed by a criticality level down.

In case we modify the characteristics of Task 3 to $\vec{C}_3(\Lambda_3) = \vec{C}_3(\Lambda_2) = 24$ ms, we even have a criticality level up from Λ_3 to Λ_1 at time $t = 47$ would immediately be followed by a criticality level down to Λ_3. This behavior is clearly undesirable.

7.2. Improving AMC*

To prevent (or at least mitigate) the undesirable behavior identified in the previous section, we propose to bound the time provided to a task τ_i at its representative criticality level λ_i to its worst-case computation time $\vec{C}(\lambda_i)$, e.g. through resource reservation with temporal protection [35], rather than raising the criticality level. Similar to Quality-of-Service like approaches [36,37], tasks therefore have to *get by* with a budget given by $\vec{C}(\lambda_i)$ at their representative criticality level. Hence, we propose to adapt both AMC and AMC*, and complement these schemes with resource reservation.

7.2.1. Adaption of AMC

First, we change the condition $\Lambda \leq \lambda_i < \Lambda_1$ in Condition 1 of the AMC scheme to $\Lambda < \lambda_i \leq \Lambda_1$, effectively suppressing a criticality level up upon an overrun at a task's representative criticality level, i.e. Condition 1 now becomes:

Condition 3. *When a job of task τ_i is executing at time t while the system is running at level Λ with $\Lambda < \lambda_i \leq \Lambda_1$ and the actual execution time $\beta_i(t)$ equals the worst-case computation time $\vec{C}_i(\Lambda)$ of τ_i, a criticality level up occurs.*

7.2.2. Adaption of AMC*

Next, we reconsider the three topics for our AMC*-scheme, that were discussed in Section 4. As mentioned above, an overrun of $\vec{C}_i(\lambda_i)$ is now prevented by a resource reservation. With this change, handling that overrun becomes the responsibility of the (developer of the) task and part of the specification of the task. In this way, we also resolved the unspecified behavior for an overrun of a task at a criticality level Λ_1. Upon a criticality level up change, the three cases distinguished in Section 4.2 simplify to only one case:

1. $\Lambda < \lambda_i \leq \Lambda_1$: When a task τ_i overruns its worst-case computation time at the criticality level $\Lambda < \lambda_i$, the new criticality level Λ^{new} remains unchanged if $\vec{C}_i(\Lambda) = \vec{C}_i(\lambda_i)$ and becomes the smallest criticality level not giving rise to an overrun for τ_i otherwise, i.e.

$$\begin{aligned} \Lambda^{\text{new}} = \; & \textbf{if } \vec{C}_i(\Lambda) = \vec{C}_i(\lambda_i) \\ & \textbf{then} \\ & \quad \Lambda \\ & \textbf{else} \\ & \quad \min\left\{\lambda \in \mathcal{L} | \vec{C}_i(\Lambda) < \vec{C}_i(\lambda)\right\} \\ & \textbf{fi} \end{aligned} \tag{5}$$

Note that by keeping the criticality level unchanged when $\vec{C}_i(\Lambda) = \vec{C}_i(\lambda_i)$, we prevent a criticality level up whenever the resource reservation already bounds the execution of task τ_i. The policies for criticality changes, being the third topic for our AMC*-scheme discussed in Section 4.3, remain unaltered.

7.2.3. Resource reservation

Finally, for every task $\tau_i \in \mathcal{T}$ we assume a resource reservation ρ_i with a priority equal to the priority of τ_i and a capacity $\vec{C}_i(\lambda_i)$ that is replenished when τ_i is activated and lost when τ_i becomes idle. Task τ_i can execute using the capacity of ρ_i as long as the system is executing at a criticality level Λ at most equal to τ_i's representative criticality level λ_i. At a higher criticality level, ρ_i will be disabled.

Additional policies and mechanisms to support a (developer and a) task upon detecting and/or handling an overrun are conceivable, such as means to measure progress [37], but fall outside the scope of this paper.

8. Conclusion

In this paper, we described our experience with extending an OSEK-compliant RTOS with mixed criticality support. Instead of selecting a specific RTOS, we started our investigations with ExSched [10], an operating system independent external CPU scheduler supporting multiple operating systems. For our initial experiments, we used ExSched in combination with Linux. We selected AMC [9] as a basic mixed criticality scheme, extended its model from two to multiple criticality levels, and complemented it with specified behavior for criticality level up and criticality level down functionality. Extending ExSched required both extensions and revisions of the ExSched Module. In particular, we incorporated generic functionality usable for multiple plug-ins, such as run-time monitoring based on high-resolution timers and task management services, e.g. *suspend* and *resume*, and extended the plug-in interface of the ExSched Module. In addition, we developed a dedicated plug-in for mixed criticality, baptized AMC* Plugin Module, and complemented that kernel module with an AMC* Library in user space. In particular, we used a similar design structure for AMC* as for ExSched itself, effectively increasing the modularity of ExSched. Our extension requires minimal overhead when a criticality level up occurs by postponing clean-up actions till

a criticality level down occurs. We built a working prototype of our system, as demonstrated through visualized traces using Grasp [30].

Despite the fact that ExSched is a great research vehicle, we decided to abandon ExSched based on our experiences with and insights gained during the extension of ExSched with mixed criticality support. During our subsequent investigations, we directed our attention to the OSEK-compliant RTOS μC/OS-II and its extension with RELTEQ [1]. Compared to extending ExSched with AMC*, extending μC/OS-II and RELTEQ with AMC* turned out to be straightforward. Unfortunately, our implementation required changes to μC/OS-II, whereas the implementation in ExSched required no patches (i.e. modifications) to the original source code of Linux. Although μC/OS-II has been stable for many years, new kernel versions will require updates of our system. Extending μC/OS-II with AMC* without making changes to the original source code seems considerably less straightforward than our implementation, however.

Finally, we briefly reflected on AMC and AMC*. We encountered undesirable behavior of both the original and the extended scheme, i.e. erroneous and unspecified behavior as well as a criticality level up immediately followed by a criticality level down, and described improvements for both schemes in combination with resource reservation. As future work, we aim at enhancing our implementation with the described improvement of the AMC* scheme. Additional policies and mechanisms to support (a developer and) a task upon detecting and/or handling an overrun are a topic of future work as well.

References

[1] M. Holenderski, R. Bril, and J. Lukkien, "An efficient hierarchical scheduling framework for the automotive domain," in *Real-Time Systems, Architecture, Scheduling, and Application*, D.S.M. Babamir, Ed. InTech, 2012, pp. 67–94.

[2] A. Burns and R. Davis, "Mixed criticality systems – a review," University of York, UK, Tech. Rep., 2018. [Online]. https://www-users.cs.york.ac.uk/burns/review.pdf

[3] i-GAME, *Grand Cooperative Driving Challenge 2016*, 2016. [Online]. http://www.gcdc.net/en/

[4] S. Baruah and A. Burns, "Implementing mixed criticality systems in Ada," in *Proceedings 16th Ada-Europe International Conference on Reliable Software Technologies*, 2011, pp. 174–188.

[5] J. Herman, C. Kenna, M. Mollison, J. Anderson, and D. Johnson, "RTOS support for multi-core mixed-criticality systems," in *Proceedings 18th Real-Time and Embedded technology and Applications Symposium (RTAS)*, 2012, pp. 197–208.

[6] Y.S. Kim and H.W. Jin, "Towards a practical implementation of criticality mode changes in RTOS," in *Proceedings 19th IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, 2014.

[7] A. Kritikakou, C. Pagetti, C. Rochange, M. Roy, M. Faugère, S. Girbal, and G. Pérez, "Distributed run-time WCET controller for concurrent tasks in mixed-criticality systems," in *Proceedings 22nd International Conference on Real-Time Networks and Systems*, 2014, pp. 139–148.

[8] "OSEK/VDX operating system," OSEK group, Tech. Rep., 2005. [Online]. https://web.archive.org/web/20160310151905/http://portal.osek-vdx.org/files/pdf/specs/os223.pdf

[9] S. Baruah, A. Burns, and R. Davis, "Response-time analysis for mixed criticality systems," in *Proceedings 32nd IEEE Real-Time Systems Symposium*, 2011, pp. 34–43.

[10] M. Åsberg, T. Nolte, S. Kato, and R. Rajkumar, "ExSched: An External CPU Scheduler Framework for Real-Time Systems," in *Proceedings 18th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2012, pp. 240–249.

[11] J. Calandrino, H. Leontyev, A. Block, U. Devi, and J. Anderson, "LITMUS^{RT}: A testbed for empirically comparing real-time multiprocessor schedulers," in *Proceedings 27th IEEE Real-Time Systems Symposium (RTSS)*, 2006, pp. 111–123.

[12] L. Palopoli, T. Cucinotta, L. Marzario, and G. Lipari, "AQuoSA: Adaptive quality of service architecture," *Software Practice and Experience*, Vol. 39, No. 1, 2009, pp. 1–31.

[13] *Ubuntu Kernel Repository.* [Online]. http://kernel.ubuntu.com/~kernel-ppa/mainline/

[14] J. Labrosse, *MicroC/OS-II: The Real Time Kernel*, 2nd ed. CMP Books, 2002.

[15] T. Gupta, E. Luit, M. van den Heuvel, and R. Bril, "Extending ExSched with mixed criticality support – an experience report," in *Proceedings 3rd Workshop on Automotive System/-Software Architecture (WASA), In conjunction with IEEE International Conference on Software Architecture (ICSA)*, 2017.

[16] S. Vestal, "Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance," in *Proceedings 28th IEEE Real-Time Systems Symposium (RTSS)*, 2007, pp. 239–243.

[17] S. Baruah and S. Vestal, "Schedulability analysis of sporadic tasks with multiple criticality specifications," in *Proceedings 20th Euromicro Conference on Real-Time Systems (ECRTS)*, 2008, pp. 147–155.

[18] F. Santy, L. George, P. Thierry, and J. Goossens, "Relaxing mixed-criticality scheduling strictness for task sets scheduled with FP," in *Proceedings 24th Euromicro Conference on Real-Time Systems (ECRTS)*, 2012, pp. 155–165.

[19] T. Gupta, "Extending a real-time operating system with a mechanism for criticality-level changes," Eindhoven University of Technology, Stan Ackermans Institute (SAI), Tech. Rep. 2015027, 2015. [Online]. https://pure.tue.nl/ws/files/32337443/Final_Report_Tarun_Gupta.pdf

[20] T. Gupta, "Interoperable robustness measures for safety-integrity levels (SILs)," European Commission 7th Framework Programme, i-GAME Deliverable D2.4, 2015. [Online]. http://www.gcdc.net/images/doc/D2.4.Interoperable.robustness.measures.for.safety-integrity.level.pdf

[21] A. Massa, *Embedded Software Development with eCOS*. Prentice Hall, 2003.

[22] G. Durrieu, M. Faugèere, S. Girbal, D. Garcia Pérez, C. Pagetti, and W. Puffitsch, "Predictable flight management system implementation on a multicore processor," in *Proceedings 9th Embedded Real-Time Software*, 2014.

[23] M. Holenderski, W. Cools, R. Bril, and J. Lukkien, "Extending an open-source real-time operating system with hierarchical scheduling," Eindhoven University of Technology (TU/e), Tech. Rep. CS-report 10-10, 2010.

[24] M. Holenderski, W. Cools, R.J. Bril, and J. Lukkien, "Multiplexing real-time timed events," in *Proceedings 14 IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2009.

[25] M. Heuvel, R. Bril, and J. Lukkien, "Transparent synchronization protocols for compositional real-time systems," *IEEE Transactions on Industrial Informatics*, Vol. 8, No. 2, 2012, pp. 322–336.

[26] M. Heuvel, E. Luit, R. Bril, J. Lukkien, P. Verhoeven, and M. Holenderski, "An experience report on the integration of ECU software using an HSF-enabled real-time kernel," in *Proceedings 11th Workshop on Operating Systems Platforms for Embedded Real-Time applications (OSPERT)*, 2015, pp. 51–56.

[27] C. Liu and J. Layland, "Scheduling algorithms for multiprogramming in a real-time environment," *JACM*, Vol. 20, No. 1, 1973, pp. 46–61.

[28] R. Bril, J. Lukkien, and W. Verhaegh, "Worst-case response time analysis of real-time tasks under fixed-priority scheduling with deferred preemption," *Real-Time Systems Journal*, Vol. 42, No. 1-3, 2009, pp. 63–119.

[29] S. Kato, R. Rajkumar, and Y. Ishikawa, "A loadable real-time scheduler suite for multicore platforms," Technical Report CMU-ECE-TR09-12, Tech. Rep, Tech. Rep., 2009.

[30] M. Holenderski, M. Heuvel, R. Bril, and J. Lukkien, "Grasp: Tracing, visualizing and measuring the behavior of real-time systems," in *Proceedings 1st Int. Workshop on Analysis Tools and Methodologies for Embedded and Real-Time Systems (WATERS)*, 2010, pp. 37–42.

[31] P. Gai, E. Bini, G. Lipari, M. Di Natale, and L. Abeni, "Architecture for a portable open source real time kernel environment," in *2nd Real-Time Linux Workshop and Hand's on Real-Time Linux Tutorial*, 2000.

[32] *Master Automotive Technology*, Eindhoven Univeristy of Technology (TU/e), 2017. [Online]. https://educationguide.tue.nl/programs/graduate-school/masters-programs/automotive-technology/

[33] *OpenRISC overview*, OpenCores, 2009. [Online]. http://www.opencores.org/project,or1k

[34] M. Behnam, T. Nolte, I. Shin, M. Åsberg, and R.J. Bril, "Towards hierarchical scheduling in VxWorks," in *Proceedings 4th International Workshop on Operating Systems Platforms for Embedded Real-Time Applications (OSPERT)*, 2008, pp. 63–72.

[35] R. Rajkumar, K. Juvva, A. Molano, and S. Oikawa, "Resource kernels: A resource-centric approach to real-time and multimedia systems," in *Proceedings SPIE, Vol. 3310, Conference on Multimedia Computing and Networking (CMCN)*, 1998, pp. 150–164.

[36] R.J. Bril and E. Steffens, "User focus in consumer terminals and conditionally guaranteed budgets," in *Proceedings 9th International Workshop on Quality of Service (IWQoS)*, ser.] Lecture Notes in Computer Science (LNCS), No. 2092, 2001, pp. 107–120.

[37] C. Wüst, L. Steffens, W. Verhaegh, R.J. Bril, and C. Hentschel, "QoS control strategies for high-quality video processing," *Real-Time Systems*, Vol. 30, No. 1–2, 2005, pp. 7–29.

e-Informatica Software Engineering Journal (eISEJ) is an international, open access, no authorship fees, blind peer-reviewed journal that concerns theoretical and practical issues pertaining development of software systems. Our aim is to focus on experimentation and machine learning in software engineering.

The journal is published under the auspices of the *Polish Academy of Sciences, Committee of Computer Science, Software Engineering Section*.

Aims and Scope:

The purpose of **e-Informatica Software Engineering Journal** is to publish original and significant results in all areas of software engineering research.

The scope of **e-Informatica Software Engineering Journal** includes methodologies, practices, architectures, technologies and tools used in processes along the software development lifecycle, but particular stress is laid on empirical evaluation.

e-Informatica Software Engineering Journal is published online and in hard copy form. The on-line version is from the beginning published as a gratis, no authorship fees, open access journal, which means it is available at no charge to the public. The printed version of the journal is the primary (reference) one.

Topics of interest include, but are not restricted to:

— Software requirements engineering and modeling
— Software architectures and design
— Software components and reuse
— Software testing, analysis and verification
— Agile software development methodologies and practices
— Model driven development
— Software quality
— Software measurement and metrics
— Reverse engineering and software maintenance
— Empirical and experimental studies in software engineering (incl. replications)
— Evidence based software engineering
— Systematic reviews and mapping studies
— Meta-analyses
— Object-oriented software development
— Aspect-oriented software development
— Software tools, containers, frameworks and development environments
— Formal methods in software engineering.
— Internet software systems development
— Dependability of software systems
— Human-computer interaction
— AI and knowledge based software engineering
— Data mining in software engineering
— Prediction models in software engineering
— Mining software repositories
— Search-based software engineering
— Multiobjective evolutionary algorithms
— Tools for software researchers or practitioners
— Project management
— Software products and process improvement and measurement programs
— Process maturity models

Important information: Papers can be rejected administratively without undergoing review for a variety reasons, such as being out of scope, being badly presented to such an extent as to prevent review, missing some fundamental components of research such as the articulation of a research problem, a clear statement of the contribution and research methods via a **structured abstract** or the evaluation of the proposed solution (empirical evaluation is strongly suggested).

Funding acknowledgements: Authors are requested to identify who provided financial support for the conduct of the research and/or preparation of the article and to briefly describe the role of the sponsor(s), if any, in study design; in the collection, analysis and interpretation of data; in the writing of the paper. If the funding source(s) had no such involvement then this should be stated as well.

The submissions will be accepted for publication on the base of positive reviews done by international Editorial Board (`http://www.e-informatyka.pl/index.php/einformatica/editorial-board/`) and external reviewers. English is the only accepted publication language. To submit an article please enter our online paper submission site (`https://mc.manuscriptcentral.com/e-InformaticaSEJ`).

Subsequent issues of the journal will appear continuously according to the reviewed and accepted submissions.

http://www.e-informatyka.pl/

e-Informatica

ISSN 1897-7979