e-Informatica Software Engineering Journal, Volume 3, Issue 1, 2009

Satisfying Stakeholders' Needs – Balancing Agile and Formal Usability Test Results

Jeff Winter*, Kari Rönkkö*

*School of Engineering, Dept. of Interaction & System Design, Blekinge Institute of Technology

jeff.winter@bth.se, kari.ronkko@bth.se

Abstract

This paper deals with a case study of testing with a usability testing package (UTUM), which is also a tool for quality assurance, developed in cooperation between industry and research. It shows that within the studied company, there is a need to balance agility and formalism when producing and presenting results of usability testing to groups who we have called Designers and Product Owners. We have found that these groups have different needs, which can be placed on opposite sides of a scale, based on the agile manifesto. This becomes a Designer and a Product Owner Manifesto. The test package is seen as a successful hybrid method combining agility with formalism, satisfying organisational needs, and fulfilling the desire to create a closer relation between industry and research.

1. Introduction

Product quality is becoming the dominant success criterion in the software industry, and Osterweil states that the challenge for research is to provide the industry with the means to deploy quality software, allowing companies to compete effectively [23]. Quality is multi-dimensional, and impossible to show through one simple measure, and research should focus on identifying various dimensions of quality and measures appropriate for it [23]. A more effective collaboration between practitioners and researchers would be of great value [23]. Quality is also important owing to the criticality of software systems (a view supported by Harrold in her roadmap for testing [14]) and even to changes in legislation that make executives responsible for damages caused by faulty software.

One approach to achieving quality has been to rely on complete, testable and consistent requirements, traceability to design, code and test cases, and heavyweight documentation. However, a demand for continuous and rapid results in a world of continuously changing business decisions often makes this approach impractical or impossible, pointing to a need for agility. At a keynote speech at the 5th Workshop on Software Quality, held at ICSE 2007 [45], Boehm stated that both agility and quality are becoming more and more important. Many areas of technology exhibit a tremendous pace of change, due to changes in technology and related infrastructures, the dynamics of the marketplace and competition, and organisational change. This is particularly obvious in mobile phone development, where their pace of development and penetration into the market has exploded over the last 5 years. This kind of situation demands an agile approach [6].

This article is based on two case studies of a usability evaluation framework called UIQ Technology Usability Metrics (UTUM) [39], the result of a long research cooperation between the research group "Use-Oriented Design and Development" (U-ODD) [37] at Blekinge Institute of Technology (BTH), and UIQ Technology (UIQ) [38]. With the help of Martin et al.'s study [21]

and our own case studies, it presents an approach to achieving quality, related to an organizational need for agile and formal usability test results. We use concepts such as "agility understood as good organizational reasons" and "plan driven processes as the formal side in testing", to identify and exemplify a practical solution to assuring quality through an agile approach. The research question for the first case study was:

– How can we balance demands for agile results with demands for formal results when performing usability testing for quality assurance?

We use the term "formal" as a contrast to the term "agile" not because we see agile processes as being informal or unstructured, but since "formal" is more representative than "plan driven" to characterise the results of testing and how they are presented to certain stakeholders. We examine how the results of the UTUM test are suitable for use in an agile process. eXtreme Programming (XP) is used as an illustrative example in this article, but note that there is no strong connection to any particular agile methodology; rather, there is a philosophical connection between the test and the ideas behind the agile movement. We examine how the test satisfies requirements for formal and informal statements of usability and quality.

In the first study, we identify two groups of stakeholders that we designated as Designers (D) and Product Owners (PO), with an interest in the different elements of the test data. A further case study was performed to discover if these findings could be confirmed. It attempted to answer the following research questions:

– Are there any presentation methods that are generally preferred?
– Is it possible to find factors in the data that allow us to identify differences between the separate groups (D & PO) that were tentatively identified in the case study presented in the previous chapter?
– Are there methods that the respondents think are lacking in the presentation methods currently in use within UTUM?
– Do the information needs, and preferred methods change during different phases of a design and development project?
– Can results be presented in a meaningful way without the test leader being present?

The structure of the article is as follows. An overview of two testing paradigms is provided. A description of the test method comes next, followed by a presentation of the methodology, and the material from the case studies, examining the balance between agility and formalism, the information needs of different stakeholders, the relationship between agility, formality and quality, and the need for research/industry cooperation. The article ends with a discussion of the work, and conclusions.

2. Testing – Prevailing Models vs. Agile Testing

Testing is performed to support quality assurance, and an emphasis on software quality requires improved testing methodologies that can be used by practitioners to test their software [14]. Since we regard the test framework as an agile testing methodology, this section presents a discussion of testing from the viewpoints of both the software engineering community and the agile community.

Within software engineering, there are many types of testing, in many process models, (e.g. the Waterfall model [30], Boehm's Spiral model [4]). Testing is often phase based, and the typical stages of testing (see e.g. [33], [25]) are *Unit testing, Integration testing, Function testing, Performance testing, Acceptance testing, and Installation testing*. The stages from Function testing and onwards are characterised as *System Testing*, where the system is tested as a whole rather than as individual pieces [25]. Usability testing (otherwise named Human Factors Testing) has been characterised as investigating requirements dealing with the user interface, and has been regarded as a part of Performance testing [25]. The prevailing approach to testing is reliant on formal aspects and best practice.

Agile software development changes how software development organisations work, especially regarding testing [34]. In agile development, exemplified here by XP [1], a key tenet is that testing is performed continuously by developers. Tests should be isolated, i.e. should not interact with the other tests that are written, and should preferably be automatic, although not all companies applying XP automate all tests [21]. Tests come from both programmers and customers, who create tests that serve to increase their confidence in the operation of the program. Customers specify functional tests to show that the system works how they expect it to, and developers write unit tests to ensure that the programs work how they think it does. These are the main testing methods in XP, but can be complemented by other types of tests when necessary. Some XP teams may have dedicated testers, who help customers translate their test needs into tests, who can help customers create tools to write, run and maintain their own tests, and who translate the customer's testing ideas into automatic, isolated tests [1].

The role of the tester is a matter of debate. It is primarily developers who design and perform testing. However, within industry, there are seen to be fundamental differences between the people who are "good" testers and those who are good developers. In theory, it is often assumed that the tester is also a developer, even when teams use dedicated testers. Within industry, however, it is common that the roles are clearly separated, and that testers are generalists with the kind of knowledge that users have, who complement the perspectives and skills of the testers. A good tester can have traits that are in direct contrast with the traits that good developers need (see e.g. Pettichord [24] for a discussion regarding this). Pettichord claims that good testers think empirically in terms of observed behaviour, and must be encouraged to understand customers' needs. Thus, although there are similarities, there are substantial differences in testing paradigms, how they treat testing, and the role of the tester and test designer. In our testing, the test leaders are specialists in the area of usability and testing, and generalists in the area of the product and process as a whole.

3. The UTUM Usability Evaluation Framework

UTUM is a usability evaluation framework for mass market mobile devices, and is a tool for quality assurance, measuring usability empirically on the basis of metrics for satisfaction, efficiency and effectiveness, complemented by a test leader's observations. Its primary aim is to measure usability, based on the definition in ISO 9241-11, where usability is defined as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" [17]. This is similar to the definition of quality in use defined in ISO 9126-1, where usability is instead defined as understandability, learnability and operability [18]. The intention of the test is also to measure "The User eXperience" (UX), which is seen as more encompassing than the view of usability that is contained in e.g. the ISO standards [39], although it is still uncertain how UX differs from the traditional usability perspective [41] and exactly how UX should be defined (for some definitions, see e.g. ([15], [16], [42]).

In UTUM testing, one or more test leaders carry out the test according to predefined requirements and procedure. The test itself takes place in a neutral environment rather than a lab, in order to put the test participant at ease. The test is led by a test leader, and it is performed together with one tester at a time. The test leader welcomes the tester, and the process begins with the collection of some data regarding the tester and his or her current phone and typical phone use. Whilst the test leader is preparing the test, the tester has the opportunity to get acquainted with the device to be tested, and after a few minutes is asked to fill in a hardware evaluation, a questionnaire regarding attitudes to the look and feel of the device.

The tester performs a number of use cases on the device, based on the tester's normal phone

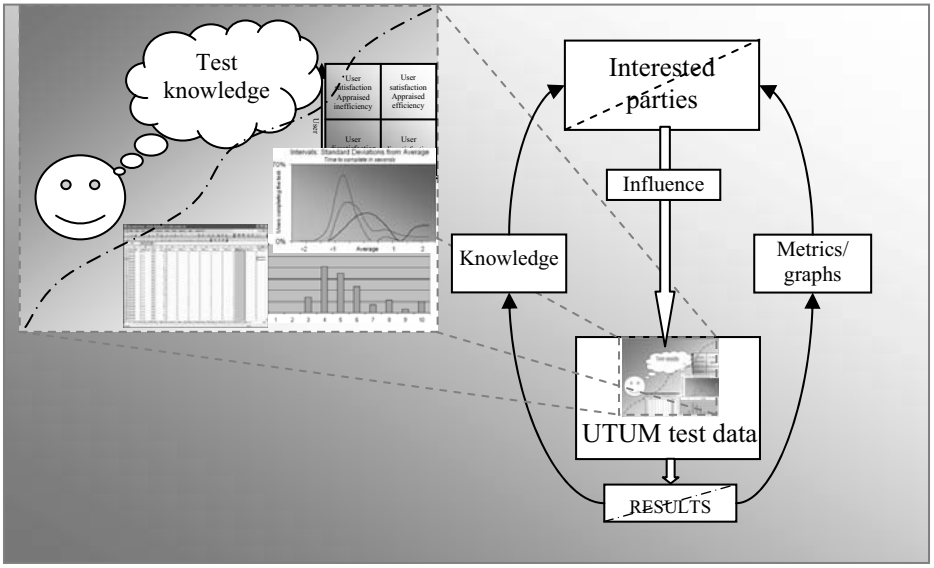

Figure 1. Contents of the UTUM testing, a mix of metrics and mental data

use or organisational testing needs. The test leader observes what happens during the use case performance, and records any observations, the time taken to complete the use cases, and answers to follow-up questions that arise. After the use case is complete, the tester answers questions about how well the telephone lets the user accomplish the use case.

When all of the use cases are completed, the tester completes a questionnaire based on the System Usability Scale (SUS) [7] about his or her subjective impressions of how easy the interface is to use. It expresses the tester's opinion of the phone as a whole. The tester is finally thanked for their participation in the test, and is usually given a small gift, such as a cinema ticket, to thank them for their help.

The data obtained are transferred to spreadsheets. These contain both quantitative data, such as use case completion times and attitude assessments, and qualitative data, such as comments made by testers and information about problems that arose. The data is used to calculate metrics for performance, efficiency, effectiveness and satisfaction, and the relationships between them, leading to a view of usability for the device as a whole. The test leader is an important source of data and information in this process, as he or she has detailed knowledge of what happened during testing.

Figure 1 illustrates the flow of data and knowledge contained in the test and the test results, and how the test is related to different groups of stakeholders. Stakeholders, who can be within the organisation, or licensees, or customers in other organisations, can be seen at the top of the flow, as interested parties. Their requirements influence the design and contents of the test. The data collected is found both as knowledge stored in the mind of the test leader, and as metrics and qualitative data in spreadsheets.

The results of the testing are thereby a combination of metrics and knowledge, where the different types of data confirm one another. Metrics based material is presented in the form of diagrams, graphs and charts, showing comparisons, relations and tendencies. This can be corroborated by the knowledge possessed by the test leader, who has interacted with the testers and who knows most about the process and context of the testing. Knowledge material is often presented verbally, but can if necessary be supported and confirmed by metrics and visual presentations of the data.

UTUM has been found to be a customer driven tool that is quick and efficient, is easily transferable to new environments, and that handles complexity [44]. For more detailed information on the contents and performance of

the UTUM test and the principles behind it, see ([39], and [44]). A brief video presentation of the whole test process (6 minutes) can be found on YouTube [8].

4. The Study Methodology and the Case Studies

This work has been part of a long-term research cooperation between U-ODD and UIQ, which has centred on the development and evaluation of a usability evaluation framework (for more information, see [44], [40]). The case studies in this phase of the research cooperation were based on tests performed by together by UIQ in Ronneby, and by Sony Ericsson Mobile Development in Manchester.

The process of research cooperation is action research (AR) according to the research and method development methodology called Cooperative Method Development (CMD), see [11], [10], [12] and ([28], chapter 8) for further details. AR "involves practical problem solving which has theoretical relevance" ([22] p. 12). It involves gaining an understanding of a problem, generating and spreading practical improvement ideas, applying the ideas in a real world situation and spreading the theoretical conclusions within academia [22]. Improvement and involvement are central to AR, and its purpose is to influence or change some aspect of whatever the research has as its focus ([27] p. 215). A central aspect of AR is collaboration between researchers and those who are the focus of the research. It is often called participatory research or participatory action research ([27] p. 216). CMD is built upon guidelines that include the use of ethnomethodological and ethnographically inspired empirical research, combined with other methods if suitable. Ethnography is a research strategy taken from sociology, with foundations in anthropology [29]. It relies upon the first-hand experience of a field worker who is directly involved in the setting that is under investigation [29]. CMD focuses on shop floor development practices, taking the practitioners' perspective when evaluating the empirical research and deliberating improvements, and involving the practitioners in the improvements. This approach is inspired by a participatory design (PD) perspective. PD is an approach towards system design in which those who are expected to use the system are actively involved and play a critical role in its design. It includes stakeholders in design processes, and demands shared responsibility, participation, and a partnership between users and implementers [32].

These studies have been performed as case studies, defined by Yin as "an empirical enquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident" ([46], p. 13). Yin presents a number of criteria that are used to establish the quality of empirical social research and states that they should be applied both in the design and conduct of a case study. They deal with construct validity, internal validity, external validity and reliability ([46], pp. 35–39).

Three tactics are available to increase construct validity, which deals with establishing correct measures for the concepts being studied, and is especially problematic in case study research. These are: using multiple sources of information; ensuring a chain of evidence and; using member checking, i.e. having the key participants review the case study report. In this study, we have used many different sources of information. The data was obtained through observation, through a series of unstructured and semi-structured interviews [27], both face-to-face and via telephone, through participation in meetings between different stakeholders in the process, and from project documents and working material. The interviews have been performed with test leaders, and with staff on management level within the two companies. Interviews have been audio taped, and transcribed, and all material has been stored. The second case study involves the use of a survey. The mix of data and collection methods has given a triangulation of data that serves to validate the results that have been reached.

To ensure a chain of evidence a "study database" or research diary has been main-

tained. It collects all of the information in the study, allowing for traceability and transparency of the material, and reliability [46]. It is mainly computer based, and is an account of the study recording activities performed in the study, transcriptions of interviews and observation notes, and records of relevant documents and articles. The audio recordings are also stored digitally. The written document contains notations of thoughts concerning themes and concepts that arise when reading or writing material in the account of the study. The chain of evidence is also a part of the writing process.

The most important research collaborators in the industrial organisation have been an integral part of the study, and have been closely involved in many stages of the work. They have been available for testing thoughts and hypotheses during the study, giving opportunities for member checking. They have also been involved as co-authors when writing articles, which also means that member checking has been an integral part of the research.

Internal validity is especially important in exploratory case studies, where an investigator tries to determine whether one event leads to another. It must be possible to show that these events are causal, and that no other events have caused the change. If this is not done, then the study does not deal with threats to internal validity. Some ways of dealing with this are via pattern matching, explanation building, addressing rival explanations, and using logic models. This study has been a mix of exploratory and explanatory studies. To address the issues of internal validity in the case studies, we have used the general repertoire of data analysis as mentioned in the previous paragraph. The material in the research diary has been analysed to find emerging themes, in an editing approach that is consistent with Grounded Theory (see Robson [27] p. 458). The analysis process has affected the further rounds of questioning, narrowing down the focus, and shifting the main area of interest, opening up for the inclusion of new respondents who shed light on new aspects of the study. A further method for ensuring validity has been through discussions together with research colleagues, giving them the chance to react to the analysis and suggest and discuss alternative explanations or approaches.

External validity, knowing whether the results of a case study are generalisable outside the immediate case study, has been seen as a major hinder to doing case studies, as single case studies have been seen as a poor basis for generalisation. However, this is based on a fallacious analogy, where critics contrast the situation to survey research, where samples readily generalise to a larger population. In a case study, the investigator tries to generalise a set of results to a wider theory, but, generalisation is not automatic, and a theory must be tested by replicating the findings, in much the same way as experiments are replicated. Although Yin advises performing multiple-case studies, since the chances of doing a good case study are better than using a single-case design ([46], p. 53), this study has been performed as a single-case study and has been performed to generate theory. The case here represents a unique case ([46], p. 40), since the testing has mainly been performed within UIQ, and it is thereby the only place where it has been possible to evaluate the testing methodology in its actual context. One particular threat is in our study is therefore that most of the data comes from UIQ. Due to close proximity to UIQ, the interaction there has been frequent and informal, and everyday contacts and discussions on many topics have influenced the interviews and their analysis. Interaction with Sony Ericsson has been limited to interviews and discussions, but data from Sony Ericsson confirms what was found at UIQ. A further threat is that most of the data in the case study comes from informants who work within the usability/testing area, but once again, they come from two different organisations and corroborate one another, have been complemented by information from other stakeholders, and thus present a valid picture of industrial reality.

A threat in the second case study is the fact that only ten people have participated. This makes it difficult to draw generalisable conclusions from the results. Also, since the company is now disbanded, it is not possible to return

to the field to perform cross checking with the participants in the study. The analysis is therefore based on the knowledge we have of the conditions at the company and the context where they worked, and is supported by discussions with a people who were previously employed within the company, whom we are still in contact with. These people can however mainly be characterised as Designers, and therefore may not accurately reflect the views of Product Owners.

Thus, since this research is mainly based on a study of one company in a limited context, it is not possible to make confident claims about the external validity of the study. However, we can say that we have created theory from the study, and that readings appear to suggest that much of what we have found in this study can also be found in other similar contexts. Further work remains to see how applicable the theory is for other organisations in other or wider contexts. Extending the case study and performing a similar study in another organisation is a way of testing this theory, and further analysis may show that the case at UIQ is actually representative of the situation in other organisations.

Reliability deals with the replicability of a study, whereby a later investigator should be able to follow the same procedures as a previous investigator, and arrive at the same findings and conclusions. By ensuring reliability you minimize errors and bias in a study. One prerequisite for this is to document procedures followed in your work, and this can be done by maintaining a case study protocol to deal with the documentation problem, or the development of a case study database. The general way to ensure reliability is to conduct the study so that someone else could repeat the procedures and arrive at the same result ([46], pp. 35–39). The case study protocol is intended to guide the investigator in carrying out the data collection. It contains both the instrument and the procedures and general rules for data collection. It should contain an overview of the project, the field procedures, case study questions, and a guide for the case study report ([46], p. 69). As mentioned previously, a case study database has been maintained, containing the most important details of the data collection and analysis process. This ensures that the study is theoretically replicable. One problem regarding the replicability of this study, however, is that the rapidly changing conditions for the branch that we have studied mean that the context is constantly changing, whereby it is difficult to replicate the exact context of the study.

In the following, we begin by presenting the results of the first case study, and discuss in which way the results are agile or plan-driven/formal, who is interested in the different types of results, and which of the organisational stakeholders needs agile or formal results.

5. Agile or Formal?

The first focus of the study was the fact that testing was distributed, and the effect this had on the testing and the analysis of the results During the case study, as often happens in case studies [46], the research question changed. Gradually, another area of interest became the elements of agility in the test, and the balance between the formal and informal parts of the testing. The framework has always been regarded as a tool for quality, and verifying this was one purpose of the testing that this case study was based on. Given the need for agility mentioned above, the intention became to see how the test is related to agile processes and whether the items in the agile manifesto can be identified in the results from the test framework. The following is the result of having studied the material from the case study from the perspective of the spectrum of different items that are taken up in the agile manifesto.

The agile movement is based on core values, described in the agile manifesto [35], and explicated in the agile principles [36]. The agile manifesto states that: "We are uncovering better ways of developing software by doing it and by helping others do it. Through this work we have come to value: *Individuals and interactions* over processes and tools, *Working software* over comprehensive documentation, *Customer collaboration* over contract negotiation, and *Responding*

to change over following a plan. That is, while there is value in the items on the right, we value the items on the left more". Cockburn stresses that the intention is not to demolish the house of software development, represented here by the items on the right (e.g. working software over *comprehensive documentation*), but claims that those who embrace the items on the left rather than those on the right are more likely to succeed in the long run [9]. Even within the agile community there is some disagreement about the choices, but it is accepted that discussions can lead to constructive criticism. Our analysis showed that all these elements could be identified in the test and its results.

In our research we have always been conscious of a division of roles within the company, often expressed as "shop floor" and "management", and working with a participatory design perspective we have worked very much from the shop floor point of view. During the study, this viewpoint of separate groups emerged and crystallised, and two disparate groups became apparent. We called these groups Designers, represented by e.g. interaction designers and system and interaction architects, representing the shop floor perspective, and Product Owners, including management, product planning, and marketing, representing the management perspective.

When regarding this in light of the Agile manifesto, we began to see how different groups may have an interest in different factors of the framework and the results that it can produce, and it became a point of interest to see how these factors related to the manifesto and which of the groups, Designers (D) or Product Owners (PO), is mainly interested in each particular item in the manifesto. The case study data was analysed on the basis of these emerging thoughts. Where the groups were found to fit on the scale is marked in bold text in the paragraphs that follow. One of the items is changed from "Working software" to "Working information" as we see the information resulting from the testing process as a metaphor for the software that is produced in software development.

- **Individuals and interactions** – The testing process is dependent on the individuals who lead the test, and who actually perform the testing on the devices. The central figure here is the test leader, who functions as a pivot point in the whole process, interacting with the testers, observing and registering the data, and presenting the results. This interaction is clearly important in the long run from a PO perspective, but it is **D** who has the greatest and immediate benefit of the interaction, showing how users reacted to design decisions, that is a central part of the testing.
- **Processes and Tools** – The test is based upon a well-defined process that can be repeated to collect similar data that can be compared over a period of time. This is important for the designers, but in the short term they are more concerned with the everyday activities of design and development that they are involved in. Therefore we see this as being of greatest interest to **PO**, who can get a long-term view of the product, its development, and e.g. comparisons with competitors, based on a stable and standardised method.
- **Working information** – The test produces working information quickly. Directly after the short period of testing that is the subject of this case study, before the data was collated in the spreadsheets, the test leaders met and discussed and agreed upon their findings. They could present the most important qualitative findings to system and interaction architects within the two organisations 14 days after the testing began, and changes in the implementation were requested soon after that. An advantage of doing the testing in-house is having access to the test leaders, who can explain and clarify what has happened and the implications of it. This is obviously of primary interest to **D**.
- **Comprehensive documentation** – The documentation consists mainly of spreadsheets containing metrics and qualitative data. Metrics back up qualitative data and open up ways to present test results that can be understood without having to include contextual information. They make test re-

sults accessible for new groups. The quantitative data gives statistical confirmation of the early qualitative findings, but are regarded as most useful for PO, who want figures of the findings that have been reached. There is less pressure of time to get these results compiled, as the critical findings are already being implemented. The metrics can be subject to stringent analysis to show comparisons and correlations between different factors. In both organisations there is beginning to be a demand for Key Performance Indicators for usability, and although it is still unsure what these may consist of, it is still an indication of a trend that comes from **PO** level.

- **Customer collaboration** – in the testing procedure it is important for the testers to have easy access to individuals, to gain information about customer needs, end user patterns, etc. The whole idea of the test is to collect the information that is needed at the current time regarding the product and its development. How this is done in practice is obviously of concern to PO in the long run, but in the immediate day to day operation it is primarily of interest to **D**.
- **Contract negotiation** – On a high level it is up to PO to decide what sort of cooperation should take place between different organisations and customers, and this is not something that involves D, so this is seen as most important for **PO**.
- **Respond to change** – The test is easily adapted to changes, and is not particularly resource-intensive. If there is a need to change the format of a test, or a new test requirement turns up suddenly, it is easy to change the test without having expended extensive resources on the testing. It is also easy to do a "Light" version of a test to check a particular feature that arises in the everyday work of design, and this has happened several times at UIQ. This is the sort of thing that is a characteristic of the day to day work with interaction design, and is nothing that is of immediate concern for PO, so this is seen as **D**.
- **Following a plan** – From a short-term perspective, this is important for D, but since they work in a rapidly changing situation, it is more important for them to be able to respond to change. This is however important for **PO** who are responsible for well functioning strategies and long-term operations in the company.

5.1. On Opposite Sides of the Spectrum

In this analysis, we found that "Designers", as in the agile manifesto, are interested in the items on the left, rather than the items on the right (see Figure 2). We see this as being "A Designer's Manifesto". "Product Owners" are more interested in the items on the right. Boehm characterised the items on the right side as being "An Auditor Manifesto"[6]. We see it as being "A Product Owner's Manifesto". This is of course a sliding scale; some of the groups may be closer to the middle of the scale. Neither of the two groups is uninterested in what is happening at the opposite end of the spectrum, but as in the agile manifesto, while there is value in the items on one side, they value the items on the other side more. We are conscious of the fact that these two groups are very coarsely drawn, and that some groups and roles will lie between these extremes. We are unsure exactly which roles in the development process belong to which group, but are interested in looking at these extremes to see their information requirements in regard to the results of usability testing. On closer inspection it may be found that none of the groups is on the far side of the spectrum for all of the points in the manifesto. To gain further information regarding this, a case study has been performed, which we present in the next section.

6. Follow-up Study of Preferred Presentation Methods

This study is thus an investigation of attitudes regarding which types of usability findings different stakeholders need to see, and their pre-

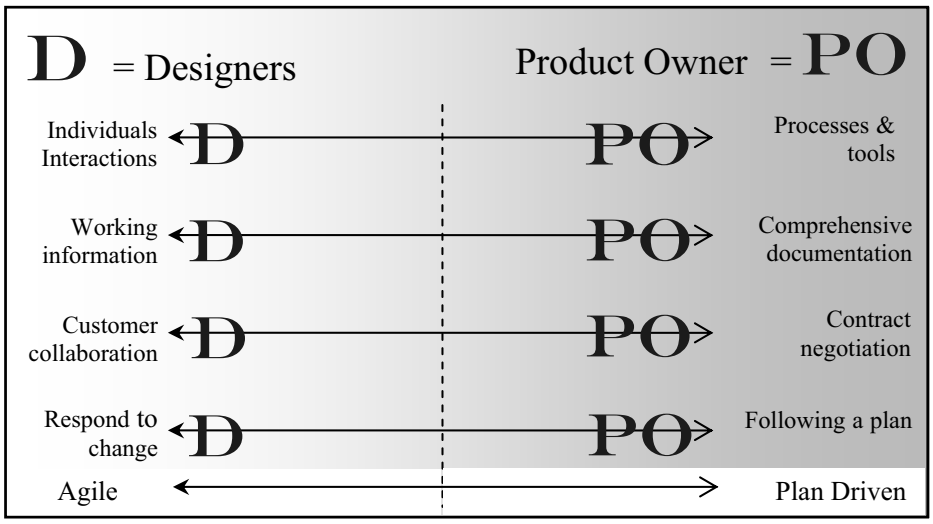

Figure 2. Groups and their diverging interests

ferred presentation methods. In the previous study we identified two groups of stakeholders with different information needs, ranging from Designers, who appear to want quick results, often qualitative results rather than quantitative results, to Product Owners, who want more detailed information, are more concerned with quantitative results, but are not as concerned with the speediness of the results. To test this theory, we sent a questionnaire to a number of stakeholders within UIQ and their customers, who are participants in the design and development process.

A document was compiled illustrating ten methods for presenting the results of UTUM tests. It contained a brief description of the presentation method and the information contained in it. The methods were chosen together with a usability expert from UIQ who often presents the results of testing to different groups of stakeholders. The methods were chosen on the basis of his experience of presenting test results to different stakeholders and are the most used and most representative ways of presenting results. The methods range from a verbal presentation of early findings, to spreadsheets containing all of the quantitative or the qualitative data from the testing, plus a number of graphical representations of the data. The methods were as follows

Method 1: The Structured Data Summary (the SDS). A spreadsheet with the qualitative findings of the testing. It shows issues that have been found, on the basis of each tester and each device, for every use case. Comments made by the test participants and observations made by the test leader are stored in the spreadsheet.

Method 2: A spreadsheet containing all "raw" data. All of the quantitative data from a series of tests. Worksheets contain the numerical data collected in a specific series of tests, which are also illustrated in a number of graphs. The data includes times taken to complete use cases, and the results of attitude assessments.

Method 3: A Curve diagram. A graph illustrating a comparison of time taken to complete one particular use case. One curve illustrates the average time for all tested telephones, and the other curves show the time taken for individual phones.

Method 4: Comparison of two factors (basic version). An image showing the results of a series of tests, where three telephones are rated and compared with regard to satisfaction and efficiency. No more information is given in this diagram.

Method 5: Comparison of two factors (brief details). The same image as Method 4, with a very brief explanation of the findings.

Method 6: Comparison of two factors (more in depth details). The same image as

Methods 4 and 5. Here, there is a more extensive explanation of the results, and the findings made by the test leader. The test leader has also written suggestions for short term and long term solutions to issues that have been found.

Method 7: The "Form Factor" – an immediate response. A visual comparison of which telephone was preferred by men and women, where the participants were asked to give an immediate response to the phones, and choose a favourite phone on the basis of "Form Factor" – the "pleasingness" of the design.

Method 8: PowerPoint presentation, no verbal presentation. A PowerPoint presentation, produced by the test leader. A summary of the main results is presented graphically and briefly in writing. This does not give the opportunity to ask follow-up questions in direct connection with the presentation.

Method 9: Verbal presentation supported by PowerPoint. A PowerPoint presentation, given by the test leader. A summary of the main results is presented graphically and briefly in writing, and explained verbally, giving the listener the chance to ask questions about e.g. the findings and suggestions for improvements. This type of presentation takes the longest to prepare and deliver.

Method 10: Verbal presentation of early results. The test leader gives a verbal presentation of the results of a series of tests. These are based mainly on his or her impressions of issues found, rather than an analysis of the metrics, and can be given after having observed a relatively small number of tests. This is the fastest and most informal type of presentation, and can be given early in the testing process.

The participants in the study were chosen together with the usability expert at UIQ. Some of the participants were people who are regularly given presentations of test results, whilst others were people who are not usually recipients of the results, but who in their professional roles could be assumed to have an interest in the results of usability testing. They were asked to read the document and complete the task by filling in their preferences in a spreadsheet. The results of the survey were returned to the researcher via e-mail, and have been summarised in a spreadsheet and then analysed on the basis of a number of criteria to see what general conclusions can be drawn from the answers.

The method whereby the participants were asked to prioritize the presentation methods was based on cumulative voting [19], [43], a well known voting system in the political and the corporate sphere ([13], [31]), also known as the $100 test or $100 method [20]. Cumulative voting is a method that has previously been used in the software engineering context, for e.g. software requirement prioritization [26] and the prioritization of process improvements [3], and in [2] where it is compared to and found to be superior to Analytical Hierarchy Process in several respects.

The questionnaire was sent to 29 people, mostly within UIQ but also to some people from UIQ's licensees. Only six respondents had replied to the questionnaire within the stipulated time, so one day after the first deadline, we sent out a reminder to the respondents who had not answered. This resulted in a further three replies. After one more week, we sent out a final reminder, leading to one more reply. Thus, we received 10 replies to the questionnaire, of which nine were from respondents within UIQ. On further enquiry, the reason given for not replying to the questionnaire was in general the fact that the company was in an intensive working phase for a planned product release, and that the staff at the company could not prioritise allocating the time needed to complete the questionnaire. This makes it impossible to give full answers to the research questions in this study, although it helps us to answer some of the questions, and gives us a better understanding of factors that affect the answers to the other questions. This study helps us formulate hypotheses for further work regarding these questions.

The division of roles amongst the respondents, and the number of respondents in the categories was as follows:

- 2: UI designers
- 2: Product planning
- 4: System design
- 1: Other (Usability)
- 1: Other (CTO Office)

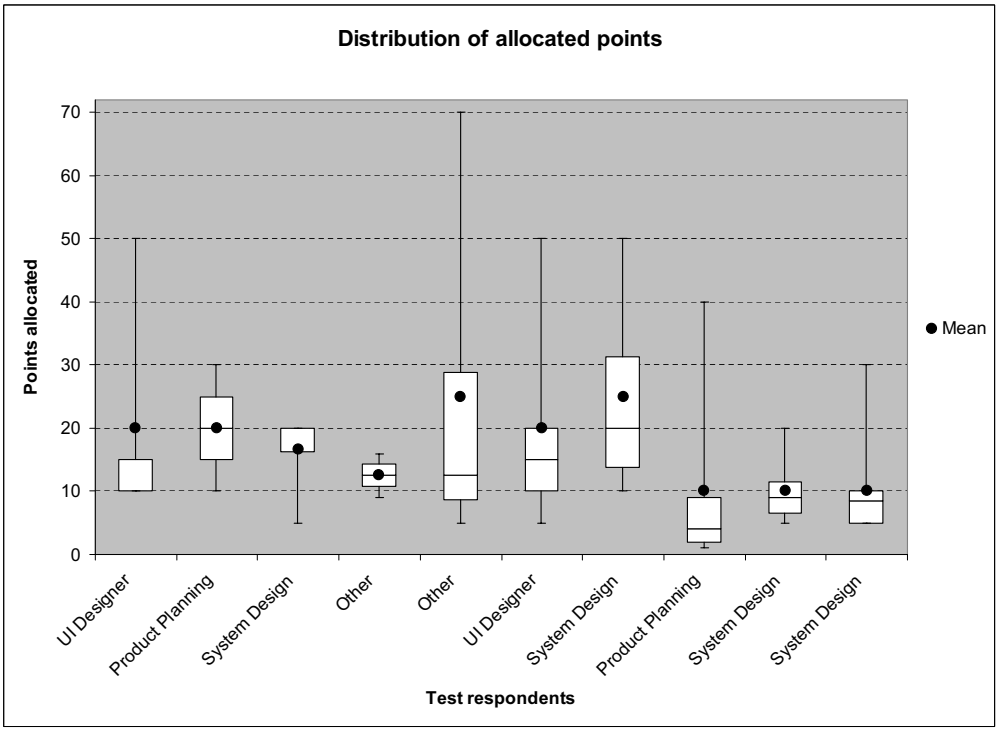

Figure 3. Distribution of points allocated per respondent

We have divided the respondents according to the tentative schema found in the first case study, between Designers (D) and Product Owners (PO). Some respondents were difficult to place in a particular category. The roles the respondents held in the company were discussed with a member of the management staff at UIQ, with long work experience at the company, who was well versed in the thoughts we had regarding the difference between Designers and Product Owners. Due to turbulence within the company, it was not possible to verify the respondents' attitudes to their positions, and would have been difficult, since they were not familiar with the terminology that we used, and the meaning of the roles that we had specified.

Five respondents, the two UI designers, the usability specialist and two of the system designers, belonged to the Designer group. The remaining five respondents, the two members of product planning, the respondent from the CTO office and two of the system designers, were representatives of the group of Product Owners.

Figure 3 is a box and whisker plot that shows the distribution of the points and the mean points allocated per person. As can be seen, the spread of points differs greatly from person to person. Although this reflects the actual needs of the respondent, the way of allocating points could also reflect tactical choices, or even the respondent's character. To get more information about how the choices were made would require a further study, where the respondents were interviewed concerning their strategies and choices.

In what follows, we use various ways of summarising the data. To obtain a composite picture of the respondents' attitudes, the methods are ranked according to a number of criteria. Given the small numbers of respondents in the study, this compilation of results is used to give a more complex picture of the results, rather than simply relying on one aspect of the questionnaire. The methods are ranked according to: the total number of points that were allocated by all respondents; the number of times the method has been chosen, and; the average ranking, which is the sum of the rankings given by each respondent, divided by the number of respondents that chose the method (e.g., if one respondent chose a method in first place, whilst another respondent chose it in third place, the average position is

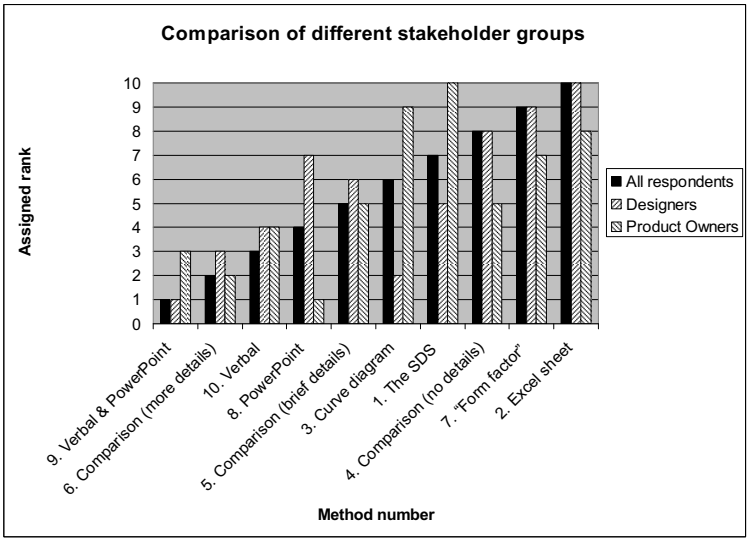

Figure 4. Comparison: All, Designers and Product Owners (lowest point is best)

$(1+3)/2 = 2$). A lower average ranking means a better result in the evaluation, although it gives no information about the number of times it has been chosen.

Figure 4 shows a summary of the results for all respondents and a comparison with the results for the group of Designers and Product Owners. For all respondents, total rank is very similar to ranking according to points allocated, and only two methods (ranked 5 and 6) have swapped places. Three methods head the list. Two are verbal presentations, one being supported by PowerPoint and the other is purely verbal.

Even within the two groups of Designers and Product Owners, there is little discrepancy between the results for total rank and position according to points awarded. Table 1 illustrates the ranks. The Methods are ordered according to the points allocated by all respondents. The next columns show the composite results, for all respondents and according to the two groups. Cases where the opinions differ significantly between Designers and Product Owners (a difference of 3 places or more) will be the subject of a brief discussion, to see whether we can draw any tentative conclusions about the presentation requirements of the different stakeholder groups. These methods, which are shown in italics in Table 1, are Methods 1, 3, 4 and 8. Since the company has now ceased operations, it is no longer possible to do a follow-up study of the attitudes of the participants, so the analysis is based on the knowledge we have of the operations at the company and the context where they worked. To verify these results, further studies are needed.

Method 3: The Curve Diagram. Designers ranked this presentation highly because if it is interpreted properly, it can give a great deal of information about the use case as it is performed on the device. If the device performs poorly in comparison to the other devices, which can easily be seen by the placement and shape of the curve, this indicates that there are problems that need to be investigated further. Use case performance time indicates the performance of the device, which can be correlated with user satisfaction. The shape of the curve illustrates when problems arose. If problems arise when performing the use case, these will be visible in the diagram and the Designers will know that there are issues that must be attended to.

Product Owners ranked this method poorly because the information is on the level of an individual use case, whilst they need information about the product or device at a coarser level of detail that is easy to interpret, giving an overall view of the product. They trust that problems at this level of detail are dealt with by the Designers, whilst they have responsibility for the product and process as a whole.

Table 1. Comparison of ranks: All, Designers and Product Owners

	Rankings			
Method	All respondents	Designers	Product Owners	Difference between groups
9. Verbal & PowerPoint	1	1	3	2
6. Comparison (more details)	2	3	2	1
10. Verbal	3	4	4	0
8. PowerPoint	4	7	1	6
5. Comparison (brief details)	6	6	5	1
3. Curve diagram	5	2	9	7
1. The SDS	7	5	10	5
4. Comparison (no details)	8	8	5	3
7. "Form factor"	9	9	7	2
2. Spreadsheet	10	10	8	2

Method 8: PowerPoint presentation, no verbal presentation. This can contain several ways of presenting the results of testing. Designers find this type of presentation of limited use because of the lack of contextual information and the lack of opportunity to pose follow-up questions. It gives a lot of information, but does not contain sufficient details to allow Designers to identify the problems or make decisions about solutions. Without details of the context and what happened in the testing situation, it is hard to interpret differences between devices, to know which problems there are in the device, and thereby difficult to know what to do about the problems. The length of time taken to produce the presentation also means that it is not suitable for Designers, who are concerned with fixing product issues as early in the development process as possible. We also believe that there is also a difference in "culture" where Designers are still unused to being presented with results in this fashion, and cannot translate this easily to fit in with their work practices.

This type of presentation is of primary interest to Product Owners because it provides an overall view of the product in comparison to other devices, without including too much information about the context and test situation. It contains sufficient text, and gives an indication of the status of the product. It is also adapted to viewing without the presence of the test leader, so the recipient can view the presentation and return to it at will. Product Owners are often schooled in an engineering tradition and are used to this way of presenting information.

Method 1: The Structured Data Summary (the SDS). Designers value this method of presentation because of the extent and character of the contextual information it includes, and because of the way the data is visualised. For every device and use case, there is information on issues that were observed, and records of comments made by the testers. It is easy to see which use cases were problematic, due to the number of comments written by the test leader, and the presence of many user comments also suggests that there are issues that need investigation. The contextual information gives clues to problems and issues that must be dealt with and gives hints on possible solutions. The effort required to read and summarise the information contained in the spreadsheet, leading to a degree of cognitive friction, means however that it is rated in the middle of the field rather than higher.

Product Owners rate this method poorly because they are uninterested in products on the level of use cases, which this presentation gives provides, and it is difficult to interpret for the device as a whole. The information is not adapted to the broad view of the product that the Product Owners need. The contextual information is difficult to summarise and does not give a readily understandable of the device as a whole. Product Owners find it difficult to make use of the information contained in this spread-

sheet and thereby rank it as least useful for their needs.

Method 4: Comparison of two factors (basic version). The lack of detail and of contextual information make if difficult for Designers to read any information that allows them to identify problems with the device. It simply provides them with a snapshot of how their product compares to other devices at a given moment.

Product Owners ranked this in the middle of the field. This is a simple way of visualising the state of the product at a given time, which is easy to compare over a period of time, to see whether a device is competitive with the other devices included in the comparison. This is typically one of the elements that are included in the PowerPoint presentation that Product Owners have ranked highest (Method 8). However, this particular method, when taken in isolation, lacks the richness of the overall picture given in Method 8 and is therefore ranked as lower.

To summarise these results, we find that the greatest difference between the two groups concerns the level of detail included in the presentation, the ease with which the information can be interpreted, and the presence of contextual information in the presentation. Designers prioritise methods that give specific information about the device and its features. Product Owners prioritise methods that give more overarching information about the product as a whole, and that is not dependent on including contextual information.

6.1. Changing Information Needs

Participants were informed that the survey was mainly focused on the presentation of results that are relevant during ongoing design and development. We pointed out that we believed that different presentation methods may be important in the starting and finishing phases of these processes. We stated that comments regarding this would be appreciated. Three respondents wrote comments about this factor.

One respondent (D) stated that the information needed in their everyday work as a UI designer, in the early stages of projects when the interaction designers are most active, was best satisfied through the verbal presentations of early results and verbal presentation supported by PowerPoint, whilst a non-verbal presentation, in conjunction with the metrics data in the spreadsheet and the SDS would be more appropriate later in the project, where the project activities were no longer as dependent on the work tasks and activities of the interaction designers.

A second respondent (D) stated that the verbal presentations are most appropriate in the requirements/design processes. Once the problem domain is understood, and the task is to iterate towards the best solution, the metrics data and the SDS would become more appropriate, because the problem is understood and the qualitative answers are more easily interpreted than the qualitative answers.

Another respondent (PO) wrote that it was important to move the focus from methods that were primarily concerned with verification towards methods that could be of assistance in requirements handling, in prioritisation and decision making in the early phases of development. In other words, the methods presented are most appropriate for later stages of a project, and there is a lack of appropriate methods for early stages.

Given the limited number of answers to these questions, it is of course difficult to draw any general conclusions, although it does appear to be the case that the verbal results are most important in the early stages of a project, to those who are involved in the actual work of designing and developing the product, whilst the more quantitative data is more useful as reference material in the later stages of a project, or further projects.

6.2. Attitudes Towards the Role of the Test Leader

The respondents were asked to judge whether or not they would need the help of the test leader in order to understand the presentation method in question. Two of the respondents supplied no answers to this question, and one of the respondents only supplied answers regarding

methods 9 and 10, which presuppose the presence of the test leader and are therefore excluded from the analysis. If we exclude these three respondents from the summary, there were seven respondents, of whom four gave answers for all eight methods, one gave five answers, and two gave three answers. The three respondents who did not answer these questions were all Product Owners, meaning that there were five designers and two Product Owners who answered these questions.

Analysis of the answers showed that, with the exception of Method 7 the methods that are primarily graphical representations of the data do not appear to require the presence of the test leader to explain the presentation. Method 7 was found to require the presence of the test leader, presumably because it was not directly concerned with the operations of the company. The spreadsheets however, one containing qualitative and one containing quantitative data, both require the presence of a test leader to explain the contents.

Given the fact that the Designers were in the majority, there were few obvious differences between Designers and Product Owners, although the most consistent findings here regard methods 4, 5, and 6, variations of the same presentation method with different amounts of written information. Here, Product Owners needed the test leader to be present whilst Designers did not.

6.2.1. In Summary

We now summarise the results of the research questions posed in this case study. The answer to the first question, whether any presentation methods are generally preferred, is that the respondents as a whole generally preferred verbal presentations. The primarily verbal methods are found in both first and third place. The most popular form was a PowerPoint presentation that was supported by verbal explanations of the findings. In second place is a non-verbal illustration showing a comparison of two factors, where detailed information is given explaining the diagram and the results it contains. This type of presentation is found in several variants in the study, and those with more explanatory detail are more popular than those with fewer details. Following these is a block of graphical presentation methods that are not designed to be dependent on verbal explanations. Amongst these is a spreadsheet containing qualitative data about the test results. At the bottom of the list is a spreadsheet that contains the quantitative data from the study. This presentation differs in character from the SDS, the spreadsheet containing qualitative data, since the SDS offers a view of the data that allows the identification of problem areas for the tested devices. This illustrates the fact that even a spreadsheet, if it offers a graphical illustration of the data that it contains, can also be found useful for stakeholders, even without an explicit explanation of the data that it contains.

Concerning the second question, we could identify differences between the two groups of stakeholders, and the greatest difference between the groups concerns the level of detail included in the presentation, the ease with which the information can be interpreted, and the presence of contextual information in the presentation. Designers prioritise methods that give specific information about the device and its features. Product Owners prioritise methods that give more overarching information about the product as a whole, and that is not dependent on including contextual information. We also found that both groups chose PowerPoint presentations as their preferred method, but that the Designers chose a presentation that was primarily verbal, whilst Product Owners preferred the purely visual presentation. Another aspect of this second question is the attitude towards the role of the test leader, where there were few obvious differences between Designers and Product Owners. The most consistent findings here concern variations of the same presentation method with different amounts of written information. Here, Product Owners needed the test leader to be present whilst Designers did not.

Regarding the third question, if there are methods that are lacking in the current presen-

tation methods, it was found that taking into account and visualising aspects of UX is becoming more important, and the results indicate that testing must be adapted to capture these aspects more implicitly. There is also a need for a composite presentation method combining the positive features of all of the current methods – however, given the fact that there do appear to be differences between information needs, it may be found to be difficult to devise one method that satisfies all groups.

No clear answers can be found for the fourth question, whether information needs, and preferred methods change during different phases of a design and development project. However, the replies suggest that the required methods do change during a project, that more verbally oriented and qualitative presentations are important in early stages of a project, in the concrete practice of design and development, and that quantitative orientated methods are important in later stages and as reference material.

Regarding the final question, whether results can be presented without the presence of the test leader, we find that the methods that are primarily graphical representations of the data do not appear to require the presence of the test leader to explain the presentation. The spreadsheets however, containing qualitative and quantitative data, both require the presence of a test leader to explain the contents.

To verify these results, further studies are of course needed. Despite the small scale of this study, the results give a basis for performing a further study, and allow us to formulate a hypothesis for following up our results. In line with the rest of the work performed as part of this research, we feel that the this work should be a survey based study in combination with an interview based study, in order to verify the results from the survey and gain a depth of information that is difficult to obtain from a purely survey based study.

We continue by discussing the results of the two case studies in relation to the industrial situation where we have been working, and the need for quality assurance in development and design processes.

7. Discussion

We begin by discussing our results in relation to academic discourses, to answer our first research question: *How can we balance demands for agile results with demands for formal results when performing usability testing for quality assurance?* We also comment upon two related discourses from the introductory chapter, i.e. the relation between quality and a need for cooperation between industry and research, and the relationship between quality and agility.

Since we work in a mass-market situation, and the system that we are looking at is too large and complex for a single customer to specify, the testing process must be flexible enough to accommodate the needs of many different stakeholders. The product must appeal to the broadest possible group, so it is difficult for customers to operate in dedicated mode with development team, with sufficient knowledge to span the whole range of the application, which is what an agile approach requires to work best [5]. In this case, test leaders work as proxies for the user in the mass market. We had a dedicated specialist test leader who brought in the knowledge that users have, in accordance with Pettichord [24]. Evidence suggests that drawing and learning from experience may be as important as taking a rational approach to testing [21]. The fact that the test leaders involved in the testing are usability experts working in the field in their everyday work activities means that they have considerable experience of their products and their field. They have specialist knowledge, gained over a period of time through interaction with end-users, customers, developers, and other parties that have an interest in the testing process and results. This is in line with the idea that agile methods get much of their agility from a reliance on tacit knowledge embodied in a team, rather than from knowledge written down in plans [5].

It would be difficult to gain acceptance of the test results within the whole organisation without the element of formalism. In sectors with large customer bases, companies require both rapid value and high assurance. This cannot be met by pure agility or plan-driven discipline; only a mix of these is sufficient, and organisations must evolve towards the mix that suits them best [5]. In our case this evolution has taken place during the whole period of the research cooperation, and has reached a phase where it has become apparent that this mix is desirable and even necessary.

In relation to the above, Osterweil [23] states that there is a body of knowledge that could do much to improve quality, but that there is "a yawning chasm separating practice from research that blocks needed improvements in both communities", thereby hindering quality. Practice is not as effective as it must be, and research suffers from a lack of validation of good ideas and redirection that result from serious use in the real world. This case study is part of a successful cooperation between research and industry, where the results enrich the work of both parts. Osterweil [23] also requests the identification of dimensions of quality and measures appropriate for it. The particular understanding of agility discussed in our case study can be an answer to this request. The agility of the test process is in accordance with the "good organisational reasons" for "bad testing" that are argued by Martin et al [21]. These authors state that testing research has concentrated mainly on improving the formal aspects of testing, such as measuring test coverage and designing tools to support testing. However, despite advances in formal and automated fault discovery and their adoption in industry, the principal approach for validation and verification appears to be demonstrating that the software is "good enough". Hence, improving formal aspects does not necessarily help to design the testing that most efficiently satisfies organisational needs and minimises the effort needed to perform testing. In the results of this work, the main reason for not adopting "best practice" is precisely to orient testing to meet organisational needs. Our case is a confirmation of [21]. Here, it is based on the dynamics of customer relationships, using limited effort in the most effective way, and the timing of software releases to the needs of customers as to which features to release. The present paper illustrates how this happens in industry, since the agile type of testing studied here is not according to "best practice" but is a complement that meets organisational needs for a mass-market product in a rapidly changing marketplace, with many different customers and end-users.

To summarise our second case study, the findings presented here are the results of a preliminary study that indicates the needs of different actors in the telecom industry. They are a validation of the ways in which UTUM results have been presented. They provide guidelines to improving the ways in which the results can be presented in the future. They are also a confirmation of the fact that there are different groups of stakeholders, the Designers and Product Owners found in our first case study, who have different information requirements. Further studies are obviously needed, but despite the small scale of this study, it is a basis for performing a wider and deeper study, and it lets us formulate a hypothesis regarding the presentation of testing results. We feel that the continuation of this work should be a survey based study in combination with an interview based study.

8. Conclusions and Further Work

In the usability evaluation framework, we have managed to implement a working balance between agility and plan driven formalism to satisfy practitioners in many roles. The industrial reality that has driven the development of this test package confirms the fact that quality and agility are vital for a company that is working in a rapidly changing environment, attempting to develop a product for a mass market. There is also an obvious need for formal data that can support the quick and agile results. The UTUM test package demonstrates one way to balance demands for agile results with demands for for-

mal results when performing usability testing for quality assurance. The test package conforms to both the Designer's manifesto, and the Product Owner's manifesto, and ensures that there is a mix of agility and formalism in the process.

The case in the present paper confirms the argumentation emphasizing 'good organizational reasons', since this type of testing is not according to "best practice" but is a complement that meets organisational needs for a mass-market product in a rapidly changing marketplace, with many different customers and end-users. This is partly an illustration of the chasm between industry and research, and partly an illustration of how agile approaches are taken to adjust to industrial reality. In relation to the former this case study is a successful cooperation between research and industry. It has been ongoing since 2001, and the work has an impact in industry, and results enrich the work of both parts. The inclusion of Sony Ericsson in this case study gave even greater possibilities to spread the benefits of the cooperative research. More and more hybrid methods are emerging, where agile and plan driven methods are combined, and success stories are beginning to emerge. We see the results of this case study and the UTUM test as being one of these success stories. How do we know that the test is successful? By seeing that it is in successful use in everyday practice in an industrial environment. We have found a successful balance between agility and formalism that works in industry and that exhibits qualities that can be of interest to both the agile and the software engineering community.

Acknowledgements This work was partly funded by The Knowledge Foundation in Sweden under a research grant for the software development project "Blekinge – Engineering Software Qualities", www.bth.se/besq. Thanks to the participants in the study and to my colleagues in the U ODD research group for their help in data analysis and structuring my writing. Thanks also to Gary Denman for permission to use the Structured Data Summary.

References

[1] K. Beck. *Extreme Programming Explained.* Addison Wesley, Reading, MA, 2000.

[2] P. Berander and P. Jönsson. Hierarchical cumulative voting (HCV) – prioritization of requirements in hierarchies. *International Journal of Software Engineering & Knowledge Engineering*, 16(6):819, 2006.

[3] P. Berander and C. Wohlin. Identification of key factors in software process management – a case study. In *2003 International Symposium on Empirical Software Engineering, ISESE '03*, pages 316–325, Rome, Italy, 2003.

[4] B. W. Boehm. A spiral model of software development and enhancement. *Computer*, 21(5):61–72, 1988.

[5] B. W. Boehm. Get ready for agile methods, with care. *Computer*, 35(1):64–69, 2002.

[6] B. W. Boehm. Keynote address, 5th Workshop on Software Quality (WoSQ), 2007.

[7] J. Brooke. System usability scale (SUS): a Quick-and-Dirty method of system evaluation user information, 1986.

[8] BTH. UIQ, usability test. http://www.youtube.com/watch?v=5IjIRlVwgeo, Aug. 2008.

[9] A. Cockburn and J. Highsmith. *Agile Software Development.* The Agile Software Development Series. Addison-Wesley, Boston, 2002.

[10] Y. Dittrich, C. Floyd, and R. Klischewski. *Doing Empirical Research in Software Engineering: finding a path between understanding, intervention and method development*, pages 243–262. MIT Press, 2002.

[11] Y. Dittrich, K. Rönkkö, J. Erickson, C. Hansson, and O. Lindeberg. Co-operative method development: Combining qualitative empirical research with method, technique and process improvement. *Journal of Empirical Software Engineering*, 2007.

[12] Y. Dittrich, K. Rönkkö, O. Lindeberg, J. Erickson, and C. Hansson. Co-operative method development revisited. *SIGSOFT Softw. Eng. Notes*, 30(4):1–3, 2005.

[13] J. N. Gordon. Institutions as relational investors: A new look at cumulative voting. *Columbia Law Review*, 94(4):124–193, 1994.

[14] M. J. Harrold. Testing: A roadmap. In *Proceedings of the Conference on the Future of Software Engineering*, Limmerick, Ireland, 2000. ACM Press.

[15] M. Hassenzahl, E. L. Law, and E. T. Hvannberg. User experience – towards a unified view. In

UX WS NordiCHI'06, pages 1–3, Oslo, Norway, 2006. cost294.org.

[16] M. Hassenzahl and N. Tractinsky. User experience – a research agenda. *Behaviour & Information Technology*, 25(2):91–97, 2006.

[17] International Organization for Standardization. ISO 9241-11 (1998): Ergonomic requirements for office work with visual display terminals (VDTs) – part 11: Guidance on usability. Technical report, 1998.

[18] International Organization for Standardization. ISO 9126-1 software engineering – product quality – part 1: Quality model, 2001.

[19] Investopedia.com. Cumulative voting. http://www.investopedia.com/terms/c/cumulativevoting.asp, Apr. 2009.

[20] D. Leffingwell and D. Widrig. *Managing Software Requirements: A Use Case Approach*, volume 2nd. Addison Wesley, 2003.

[21] D. Martin, J. Rooksby, M. Rouncefield, and I. Sommerville. 'Good' organisational reasons for 'Bad' software testing: An ethnographic study of testing in a small software company. In *ICSE '07*, Minneapolis, MN, 2007. IEEE.

[22] E. Mumford. Advice for an action researcher. *Information Technology and People*, 14(1):12–27, 2001.

[23] L. Osterweil. Strategic directions in software quality. *ACM Computing Surveys (CSUR)*, 28(4):738–750, 1996.

[24] B. Pettichord. Testers and developers think differently. *STGE magazine*, Vol. 2(Jan/Feb 2000 (Issue 1)), 2000.

[25] S. L. Pfleeger and J. M. Atlee. *Software Engineering*, volume 3rd. Prentice Hall, Upper Saddle River, NJ, 2006.

[26] B. Regnell, M. Höst, J. N. och Dag, P. Beremark, and T. Hjelm. An industrial case study on distributed prioritisation in Market-Driven requirements engineering for packaged software. *Requirements Engineering*, 6(1):51–62, 2001.

[27] C. Robson. *Real World Research*, volume 2nd. Blackwell Publishing, Oxford, 1993.

[28] K. Rönkkö. *Making Methods Work in Software Engineering: Method Deployment as a Social achievement*. PhD thesis, Blekinge Institute of Technology, School of Engineering, 2005. Dissertation Series No. 2005:04; Doctoral Thesis.

[29] K. Rönkkö. Ethnography. In P. Laplante, editor, *Encyclopedia of Software Engineering*. Taylor and Francis Group, New York, 2008.

[30] W. W. Royce. Managing the development of large software systems: concepts and techniques. In *9th international conference on Software Engineering*, pages 328–338, Monterey, California, United States, 1987. IEEE Computer Society Press.

[31] J. Sawyer and D. McRae, Jr. Game theory and cumulative voting in Illinois: 1902–1954. *The American Political Science Review*, 56(4):936–946, 1994.

[32] D. Schuler and A. Namioka. *Participatory Design – Principles and Practices*, volume 1st. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1993.

[33] I. Sommerville. *Software Engineering*, volume 8. Addison Wesley, 1982.

[34] D. Talby, O. Hazzan, Y. Dubinsky, and A. Keren. Agile software testing in a Large-Scale project. *IEEE Software*, 23(4):30–37, 2006.

[35] The Agile Alliance. The agile manifesto. http://agilemanifesto.org/, Apr. 2009.

[36] The Agile Alliance. Principles of agile software. http://www.agilemanifesto.org/principles.html, Apr. 2009.

[37] U-ODD. Use-Oriented Design and Development. http://www.bth.se/tek/u-odd, Apr. 2009.

[38] UIQ Technology. Company information. http://uiq.com/aboutus.html, June 2008.

[39] UIQ Technology. UIQ Technology Usability Metrics. http://uiq.com/utum.html, June 2008.

[40] UIQ Technology. UTUM website. http://uiq.com/utum.html, June 2008.

[41] UXEM. User eXperience Evaluation Methods in product development (UXEM). http://www.cs.tut.fi/ihte/CHI08_workshop/slides/Poster_UXEM_CHI08_V1.1.pdf, June 2008.

[42] UXNet: the user experience network. http://uxnet.org/, June 2008.

[43] Wikipedia. Cumulative voting. http://en.wikipedia.org/wiki/Cumulative_voting, Apr. 2009.

[44] J. Winter, K. Rönkkö, M. Ahlberg, M. Hinely, and M. Hellman. Developing quality through measuring usability: The UTUM test package. In *ICSE 2007*, 5th Workshop on Software Quality, at ICSE 2007, 2007.

[45] WoSQ. Fifth workshop on software quality, at ICSE 07. http://attend.it.uts.edu.au/icse2007/, June 2008.

[46] R. K. Yin and S. Robinson. *Case Study Research – Design and Methods*, volume 3rd of *Applied Social Research Methods Series*. SAGE publications, 5, 2003.