# Meeting Organisational Needs and Quality Assurance through Balancing Agile and Formal Usability Testing Results

Jeff Winter, Kari Rönkkö, Mårten Ahlberg, Jo Hotchkiss

# Meeting Organisational Needs and Quality Assurance through Balancing Agile & Formal Usability Testing Results

Jeff Winter[1], Kari Rönkkö[1], Mårten Ahlberg[2], Jo Hotchkiss[3],

[1] Blekinge Institute of Technology, SE 37050 Ronneby, Sweden
(jeff.winter, kari.ronkko)@bth.se
[2] UIQ Technology, Ronneby, Sweden
marten.ahlberg@uiq.com
[3] Sony Ericsson Mobile Communications, Warrington, England
jo.hotchkiss@sonyericsson.com

**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.

**Keywords:** agility, formalism, usability, product quality, methods & tools

## 1 Introduction

Osterweil et al [1] state that product quality is becoming the dominant success criterion in the software industry, and believe that the challenge for research is to provide the industry with the means to deploy quality software, allowing companies to compete effectively. Quality is multi-dimensional, and impossible to show through one simple measure, and they state that research should focus on identifying various dimensions of quality and measures appropriate for it and that a more effective collaboration between practitioners and researchers would be of great value. Quality is also important owing to the criticality of software systems (a view supported by Harrold in her roadmap for testing [2]) and even to changes in legislation that make executives responsible for damages caused by faulty software.

One traditional approach to quality has been to rely on complete, testable and consistent requirements, traceability to design, code and test cases, and heavyweight documentation. However, the 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 [3], 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 [4].

This paper deals with a case study of a usability test package called UIQ Technology Usability Metrics (UTUM) [5], the result of a long research cooperation between the research group "Use-Oriented Design and Development" (U-ODD) [6] at Blekinge Institute of Technology (BTH), and UIQ Technology (UIQ) [7]. With the help of Martin et al.'s study [8] and our own case study, 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 original aim of the study at hand was to examine how a distributed usability test could be performed, and the effect that the geographical separation of the test leaders had on the collection, analysis and presentation of the data. As often happens in case studies, another research question arose during the execution of the study: **How can we balance demands for agile results with demands for formal results when performing usability testing for quality assurance?**

Here, 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" in this case 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. Even though eXtreme Programming (XP) is used as an illustrative example in this article, 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 statements of usability and quality. As a result of the investigation regarding the agile and the formal, we also identify parties interested in the different elements of the test data.

Our study is in reference to Martin et al's work [8]. Our work deals with quality and the necessary balance between agility and formality from the viewpoint of "day to day organizational needs". Improving formal aspects is important, and software engineering research in general has successfully emphasized this focus. However, improving formal aspects may not help to design the testing that most efficiently satisfies organisational needs and minimises the testing effort. The main reason for not adopting "best practice" in testing is to orient testing to meet organisational needs, 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 (as is demonstrated in [8]). Both perspectives are needed!

The structure of the article is as follows. An overview of two different testing paradigms is provided. A description of the test method comes next, followed by a presentation of the study method and an analysis of the material from the case study,

examining the balance between agility and formalism, the relationship between these 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

This section presents a brief overview of testing as seen from the viewpoints of the software engineering community and the agile community. As quality becomes a dominant success factor for software, the practitioner's use of processes to support software quality will become increasingly important. Testing is one such process, performed to support quality assurance, and provide confidence in the quality of software, and an emphasis on software quality requires improved testing methodologies that can be used by practitioners to test their software [2]. This section therefore briefly discusses the field of testing, in connection with the fact that the test framework can be seen as an agile testing methodology.

Within software engineering, there are many types of testing, in many process models, (e.g. the Waterfall model [9], Boehm's Spiral model [10]). Testing has been seen as phase based, and the typical stages of testing (see e.g. [11], [12]) when developing large systems 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 [12].

Unit testing, which should be performed in a controlled environment, verifies that a component functions properly with the expected types of input. Integration testing ensures that system components work together as described in the specifications. After this testing, the system has been merged into a working system, and system testing can begin. System testing begins with Function testing, where the system is tested to ensure that it has the desired functionality, and evaluates whether the integrated system performs the functions described in the requirements specification. A performance test compares the system with the rest of the software and hardware requirements, and after the performance test, the system is regarded as being a validated system. In an acceptance test, the system is tested together with the customer, in order to check it against the customer's requirements description, to ensure that it works in accordance with customer expectations. When Acceptance testing is completed, the accepted system is installed in its proper environment, and in order to ensure that it functions as it should, an installation test is run [12]. Usability testing (otherwise named Human Factors Testing), which we are concerned with here, has been characterised as investigating requirements dealing with the user interface, and has been regarded as a part of Performance testing [12]. This is an example of the prevailing approach to testing, reliant on formal aspects and best practice.

Agile software development radically changes how software development organisations work, especially regarding testing [13]. In agile development, exemplified here by XP [14], one of the key tenets is that testing is performed continuously by developers. In XP, the tests should be isolated, i.e. should not interact with the other tests that are written, and should preferably be automatic, although a

recent study of testing practice in a small organisation has shown that not all companies applying XP automate all tests [8]. Tests come from two sources, from programmers and customers, who both create tests that serve, through continuous testing, to increase their confidence in the operation of the program. Customers write, or specify, functional tests to show that the system works in the way they expect it to, and developers write unit tests to ensure that the programs work the way they think that they work. Unit and functional tests 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 [14].

The role of the tester is a matter of debate. In both of the above paradigms it is primarily developers who design and perform testing, albeit occasionally at the request of the customer. However, within industry, there are seen to be fundamental differences between the people who are "good" testers and those who are good developers. The role of the tester as described above assumes 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 [15] 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. As can be seen in the above, although there are similarities, there are substantial differences in the 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.

There is a body of knowledge concerning usability testing, much of it in the field of Human Computer Interaction, but we have chosen not to look more closely at this. In this paper we concentrate on the studied company's organizational needs and the philosophical connection between the test and the ideas behind the agile movement.


## 3 The UTUM Test Package

UTUM is a usability test package 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" [16]. 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 [17]. 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 [5], although it is still

uncertain how UX differs from the traditional usability perspective [18] and exactly how UX should be defined (for some definitions, see e.g. ([19-21]).

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 then 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 next step is to perform a number of use cases on the device, based on the tester's normal phone use or organisational testing needs. Whilst this is taking place, the test leader observes what happens during the use case performance, and records these 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 is asked to answer questions about how well the telephone lets the user accomplish the use case.

The final step in the test, when all of the use cases are completed, is a questionnaire about the user's subjective impressions of how easy the interface is to use. This is based on the System Usability Scale (SUS) [22], and 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.

After testing, the data obtained are transferred to spreadsheets. These contain both quantitative data, such as use case completion times and attitude assessments, and qualitative data, like 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.
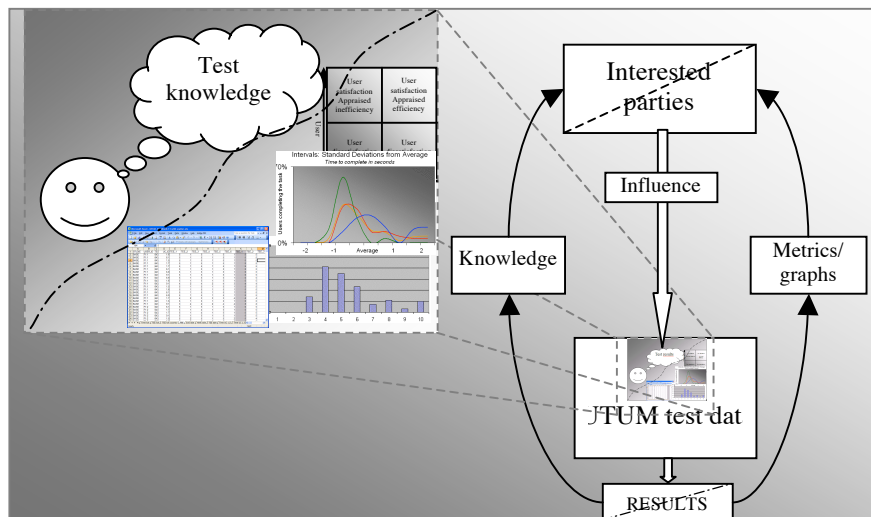


**Fig. 1.** Contents of the UTUM testing, a mix of metrics and mental data

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. Figure 2 represents a spreadsheet where qualitative findings of the testing are stored, a Structured Data Summary, created and developed by Gary Denman, UIQ. 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 as comments in the spreadsheet.

| | User | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | UserID | M1:1 | F1:1 | F3:1 | F2:1 | F1:2 | M2:1 |
| W=XXX, N=YYY, K=ZZZ | Phone Order | K:N:W | N:W:K | K:W:N | W:K:N | N:K:W | W:N:K |
| | | | | | | | |
| **XXX** | | | | | | | |
| UC1 - Receive and Answer an incoming call | | | | | | | |
| *Didn't know where to press* | | | | | | | |
| *Hesitated to press on screen* | | y | y | | | | |
| *Tried to press HW under screen* | | | | | | y | |
| *Tried to press where says 'Answer?'* | | | | | | | |
| | | | | | | | |
| UC2 - Save the incoming call as a new contact - "Joanne" | | | | | | | |
| *Save Dialog* | | y | | y | | | y |
| *Name in call log doesn't update after saving number* | | | | | y | | |
| | | | | | | | |
| UC3 - Set an alarm for 8 o'clock tomorrow morning | | | | | | | |
| *Couldn't find alarms* | | y | y | y | n | | y |
| *Puzzled by time view - where alarms* | | | | | | | |
| | | | | | | | |
| UC4 - Read an incoming SMS and reply with "I'm fine" | | | | | | | |
| | | | | | | | |
| UC5 - Make a phone call to Joanne(07949 877249) | | | | | | | |
| *Looking for Green and Red* | | | | | | | |
| | | | | | | | |
| UC6 - Create a new SMS - "Hi meet at 5" and send to Joanne | | | | | | | |
| *New SMS flow confusing - To: first* | | y | | | | | y |
| *How enter contact - when To: in focus - no "contacts"* | | y | | | | | |
| *Thought space on corp...* | | | | | | | |

**Fig. 2.** Qualitative results in spreadsheets (product information removed).

The results of the testing are thereby a combination of metrics and knowledge, where the different types of data confirm one another. The 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 is the person 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 visual presentations of the data.

UTUM has been found to be a customer driven tool that is quick and efficient, easily transferable to new environments, and that handles complexity [23]. For more detailed information on the contents and performance of the UTUM test and the principles behind it, see ([5], [23]). A brief video presentation of the whole test process (6 minutes) can be found on YouTube [24].

# 4 The Study Methodology and the Case

This work has been performed as part of a long-term cooperation between U-ODD and UIQ, which has centred on the development and evaluation of a usability test (for more information, see [23, 25]). The prime area of interest has been on creating and studying a test method for quality assurance, on developing metrics to measure usability and on the combination of qualitative and quantitative results. This case study in this phase of the research cooperation concerned tests performed by UIQ in Ronneby, and by Sony Ericsson Mobile Development in Manchester.

The process of research cooperation is Action research according to the research- and method development methodology called Cooperative Method Development (CMD), see [26] [27], [28] and ([29], chapter 8) for further details. Action research is "research that involves practical problem solving which has theoretical relevance" ([30] 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 [30]. Improvement and involvement are central to action research, and its purpose is to influence or change some aspect of whatever the research has as its focus ([31] p. 215). A central aspect of action research is collaboration between researchers and those who are the focus of the research. The terms *participatory research* and *participatory action research* are sometimes used as synonyms for action research ([31] p. 216).

CMD is built upon a number of guidelines including the use of ethnomethodological and ethnographically inspired empirical research, combined with other methods if suitable. Ethnography is a research strategy taken from sociology, and that has its foundations in anthropology[32]. It is a method that relies upon the first-hand experience of a field worker who is directly involved in the setting that is under investigation [32]. 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 designing it. It is a paradigm where stakeholders are included in the design process, and it demands shared responsibility, active participation, and a partnership between users and implementers [33].

This study has been performed as a case study, 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" ([34], s. 13). The data for the study has been obtained through observation, through a series of unstructured and semi-structured interviews [31], both face-to-face and via telephone, through participation in meetings between different stakeholders in the process, and from project documents and working material, such as a research protocol that ensures that the individual tests take place in a consistent manner, spreadsheets for storing and analysing qualitative and quantitative data, and material used for presenting results to different stakeholders. The interviews have been performed with test leaders, and with staff on management level within the two companies. Interviews have been audiotaped, and transcribed, and all material has

been collected in a research diary. The diary is also the case study database, which collects all of the information in the study, allowing for traceability and transparency of the material, and reliability [34]. The mix of data and collection methods has given a triangulation of data that serves to validate the results that have been reached.

The transcriptions of the interview material, and other case material in the research diary, have been analysed to find emerging themes, in an editing approach that is also consistent with Grounded Theory (see Robson [31] s. 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. During the case study, as often happens in case studies [34], the research question changed. 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. 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.

We have tried to counter threats to validity and reliability in the study. One of these is bias introduced by the researchers most closely involved in the study. This is addressed by cross checking results with participants in the study, and discussing the results of the case study with research colleagues. One more threat is that most of the data in the case study 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. Another 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 picture of industrial reality.

In the case in question, test leaders from two organisations in two countries performed testing in parallel. Testing was performed in a situation with complex relationships between customers, clients, and end-users, and complexities of how and where results were used. Reasons for performing the tests were to validate the UTUM test itself as a tool for quality assurance, and to obtain a greater number of tests, creating a baseline for validation of products, to identify and measure differences or similarities between countries, and to identify issues with common use-cases. Normally, there is no need for a large number of testers or data points. However, although this can be seen as a large test from the point of view of the participating organisations, compared to their normal testing, with more than 10 000 data points, it was found to be an agile process, where results were produced quickly and efficiently.

In the following, we present the results of the 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 case study was grounded in thoughts concerning the importance of quality and agility in software processes, as specified previously. We have always seen the importance of the framework 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 of this case study 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 [37] 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. Even in 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, and its connection to our test, 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 in the case study 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 material was analysed on the basis of these emerging thoughts. Where the groups were found to fit on the scale is marked at the end of the paragraphs that follow. We have changed one of the items 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 decide the format of the test, who lead the test, and who actually perform the

tests 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 is obviously 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. **D**

- **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 of interest to 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. **PO**

- **Working information** – The test produces working information quickly. Directly after the short period of testing that is the subject of this case study, the test leaders met and discussed and agreed upon their findings. This took place before the data was collated in the spreadsheets. They were able to 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 tester 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 comprehensive documentation consists of spreadsheets containing metrics and qualitative data. The increased use of metrics, which is the formal element in the testing, is seen in both organizations in this study as a complement to the testing methods already in use. Metrics back up the qualitative findings that have always been the result of testing, and open up new ways to present test results in ways that are easy to understand without having to include contextual information. They make test results 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 most important work has been done, and the critical findings are already being implemented. In this case study, the metrics consisted of 10 000 data points collected from 48 users, a mixture of quantitative measurements and attitudinal metrics. 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. **PO**

- **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 D is involved in, so this is seen as **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.
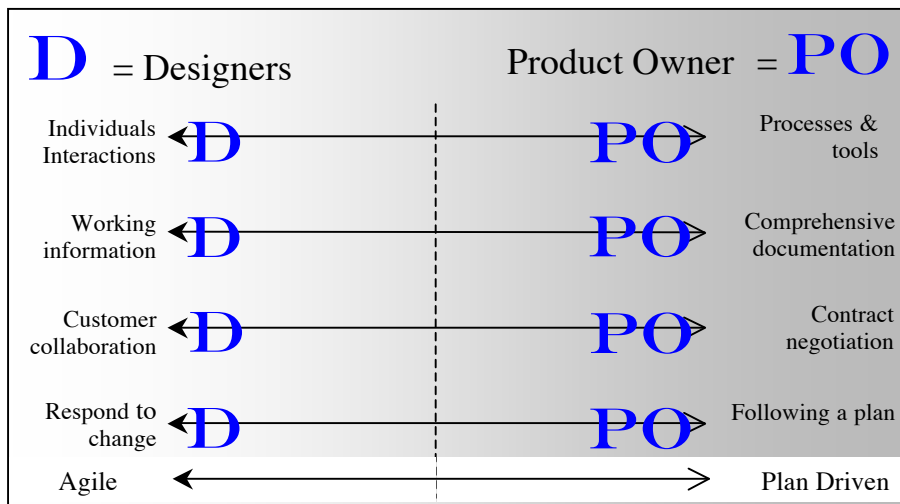
**Figure 3.** Groups and their diverging interests

**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 3). 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"[4]. 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; more work must be done to examine this distribution and division.

# 6 Discussion

Here, we discuss our results in relation to academic discourses, to answer the 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 [38]. 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 [15]. Evidence suggests that drawing and learning from experience may be as important as taking a rational approach to testing [8]. 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 [38].

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 [38]. 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 et al [1] state 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 et al

[1] also request 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 [8]. 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 the present paper, the main reason for not adopting "best practice" is precisely to orient testing to meet organisational needs. Our case is a confirmation of [8]. 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.

## 7 Conclusion and Further Work

In the UTUM test package, we have managed to implement a sufficient 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. Real-world complex situations are not either on or off. The UTUM test package demonstrates one way to balance demands for agile results with demands for formal 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 gives an even greater possibility 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.

As a follow up to this case study, work has been performed to collect more information regarding the attitudes of Product Owners and Designers towards the information they require from testing and their preferred presentation formats. This will help define the groups and their needs, and allow us to place them on the map of the manifesto, and tailor the testing and presentation methods to fulfil these needs, and thereby improve the test package even further.

## Acknowledgements

## References

1. Osterweil, L., *Strategic directions in software quality.* ACM Computing Surveys (CSUR) 1996. 28(4): p. 738 - 750.
2. Harrold, M. J. *Testing: A Roadmap.* in *Proceedings of the Conference on The Future of Software Engineering.* 2000. Limmerick, Ireland: ACM Press.
3. WoSQ. *Fifth Workshop on Software Quality, at ICSE 07.* 2007 [cited 2008-06-13]; Available from: http://attend.it.uts.edu.au/icse2007/.
4. Boehm, B. W., *Keynote address, 5th Workshop on Software Quality.* 2007: Minneapolis, MN.
5. UIQ Technology. *UIQ Technology Usability Metrics.* 2006 [cited 2008-06-13]; Available from: http://uiq.com/utum.html.
6. U-ODD. *Use-Oriented Design and Development.* 2008 [cited 2008-06-09]; Available from: http://www.bth.se/tek/u-odd.
7. UIQ Technology. *Company Information.* 2008 [cited 2008-06-12]; Available from: http://uiq.com/aboutus.html.
8. Martin, D., Rooksby, J., Rouncefield, M., Sommerville, I., *'Good' Organisational Reasons for 'Bad' Software Testing: An Ethnographic Study of Testing in a Small Software Company* in *ICSE '07.* 2007. Minneapolis, MN: IEEE.
9. Royce, W.W. *Managing the development of large software systems: concepts and techniques* in *9th international conference on Software Engineering* 1987 Monterey, California, United States IEEE Computer Society Press.
10. Boehm, B. W., *A spiral model of software development and enhancement.* Computer, 1988. 21(5): p. 61-72.
11. Sommerville, I., *Software Engineering.* 8 ed. 2007: Addison Wesley. 840.

12. Pfleeger, S. L. and J. M. Atlee, *Software Engineering*. 3rd ed. 2006, Upper Saddle River, NJ: Prentice Hall.

13. Talby, D., Hazzan, O., Dubinsky, Y., Keren, A., *Agile Software Testing in a Large-Scale Project*. IEEE Software, 2006. 23(4): p. 30-37.

14. Beck, K., *Extreme Programming Explained*. 2000, Reading, MA: Addison Wesley.

15. Pettichord, B., *Testers and Developers Think Differently*, in *STGE magazine*. 2000.

16. International Organization for Standardization, *ISO 9241-11 (1998): Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) - Part 11: Guidance on Usability*. 1998.

17. International Organization for Standardization, ISO *9126-1 Software engineering - Product quality - Part 1: Quality model*. 2001. p. 25.

18. UXEM. *User eXpreience Evaluation Methods in product development (UXEM)*. 2008 [cited 2008 2008-06-10]; Available from: http://www.cs.tut.fi/ihte/CHI08_workshop/slides/Poster_UXEM_CHI08_V1.1.pdf.

19. Hassenzahl, M., E. Lai-Chong Law, and E. T. Hvannberg. *User Experience - Towards a unified view*. in *UX WS NordiCHI'06*. 2006. Oslo, Norway: cost294.org.

20. Hassenzahl, M. and N. Tractinsky, *User experience - a research agenda*. Behaviour & Information Technology, 2006. 25(2): p. 91 - 97.

21. UXNet:. *UXNet: the User Experience network*. 2008 [cited 2008 2008-06-09]; Available from: http://uxnet.org/.

22. Brooke, J., *System Usability Scale (SUS): A Quick-and-Dirty Method of System Evaluation User Information*. 1986, Digital Equipment Co Ltd, Reading, UK

23. Winter, J., Rönkkö, K., Ahlberg, M., Hinely, M., Hellman, M., *Developing Quality through Measuring Usability – The UTUM Test Package*, in *5th Workshop on Software Quality, at ICSE 2007*. 2007, IEEE: Minneapolis, Minnesota.

24. BTH. *UIQ, Usability test*. 2008 [cited 2008-08-29]; Available from: http://www.youtube.com/watch?v=5IjIRlVwgeo.

25. UIQ Technology. *UTUM website*. 2008 [cited 2008-06-14]; Available from: http://uiq.com/utum.html.

26. Dittrich, Y., Rönkkö, K., Eriksson, J., Hansson, C., Lindeberg, O., *Co-operative Method Development: Combining qualitative empirical research with method, technique and process improvement*. Journal of Empirical Software Engineering, 2007.

27. Dittrich, Y., *Doing Empirical Research in Software Engineering – finding a path between understanding, intervention and method development*, in *Social thinking – Software practice.*, Y. Dittrich, C. Floyd, and R. Klischewski, Editors. 2002, MIT Press. p. 243-262.

28. Dittrich, Y., Rönkkö, K., Lindeberg, O., Eriksson, J, Hansson, C., *Co-Operative Method Development revisited*. SIGSOFT Softw. Eng. Notes, 2005. 30(4): p. 1-3.

29. Rönkkö, K., *Making Methods Work in Software Engineering: Method Deployment as a Social achievement*, in *School of Engineering*. 2005, Blekinge Institute of Technology: Ronneby.

30. Mumford, E., *Advice for an action researcher*. Information Technology and People, 2001. 14(1): p. 12-27.

31. Robson, C., *Real World Research*. 2nd ed. 2002, Oxford: Blackwell Publishing. 599.

32. Rönkkö, K., *Ethnography* in *Encyclopedia of Software Engineering (accepted, under review)*, P. Laplante, Editor. 2008, Taylor and Francis Group: New York.

33. Schuler, D. and A. Namioka, *Participatory Design - Principles and Practices*. 1st ed, ed. D. Schuler and A. Namioka. 1993, Hillsdale, New Jersey: Lawrence Erlbaum Associates. 319.

34. Yin, R. K., *Case Study Research - Design and Methods*. 3rd ed. Applied Social Research Methods Series, ed. S. Robinson. Vol. 5. 2003, Thousand Oaks: SAGE publications. 181.

35. The Agile Alliance, *The Agile Manifesto*. 2001 [cited 2008-06-04]; Available from: http://agilemanifesto.org/.

36. The Agile Alliance, Principles *of Agile Software*. 2001 [cited 2008-06-12]; Available from: http://www.agilemanifesto.org/principles.html.
37. Cockburn, A., *Agile Software Development*. The Agile Software Development Series, ed. A. Cockburn and J. Highsmith. 2002, Boston: Addison-Wesley.
38. Boehm, B., *Get Ready for Agile Methods, with Care.* Computer, 2002. 35(1): p. 64-69.