Measuring Usability - Balancing Agility and Formality

For Stekeholders' Needs in Software Development

Jeff Winter

Blekinge Institute of Technology Licentiate Dissertation Series
No 2009:05

Measuring Usability - Balancing Agility and Formality

For Stakeholders' Needs in Software Development

Jeff Winter

Department of Interaction and System Design
School of Engineering
Blekinge Institute of Technology
SWEDEN

Department of Interaction and System Design
School of Engineering
Publisher: Blekinge Institute of Technology
Printed by Printfabriken, Karlskrona, Sweden 2009
ISBN 978-91-7295-162-4
Blekinge Institute of Technology Licentiate Dissertation Series
ISSN 1650-2140
urn:nbn:se:bth-00438

To my ever loving and ever patient family

Abstract

The main focus of the research presented in this thesis is a usability evaluation framework for mass market mobile devices, allowing measurement, comparison and presentation of the usability of hand held devices. The research has been a cooperation between an academic and an industrial partner, based on an action research approach, following the processes of Cooperative Method Development (CMD). It has used a case study approach, where the data has been analysed using techniques taken from grounded theory. Ethnography and participatory design have been central to both the research and the cooperation.

With its basis in the evaluation framework, and its use of quantitative measurements and qualitative judgments framed by a focus on usability testing, this thesis contributes to aspects related to capturing real world usage in a continuously changing society, and supporting information needs in the process of building software.

The evaluation framework can be used as a quality assurance tool in a wide perspective. It measures usability and the user experience, quickly and flexibly, and in so doing measures aspects of quality in use. It has been tested in a complex industrial development project and is a valuable and flexible tool that is easy for a usability expert to learn and use, to measure and help build quality on the customer's terms. The results we have arrived at are of practical and theoretical interest within software engineering and the industrial telecom sphere.

Several features and aspects of the evaluation framework are new and challenging. These are: its mix of qualitative and quantitative methods, allowing it to target many different stakeholders; experience in applying these methods in the technology-focused and rapidly changing mobile phone area; the challenge of addressing end users in a mass market; the challenge of finding presentation models for the many different stakeholders; the challenge of making sure the framework can be used in different stages in the industrial software development process.

The framework is related to three areas of software engineering, Market Driven Requirements Engineering, Statistical Usage Testing, and Organisation and Product Together. Our work includes a discussion of the need for agility, which has not so far been focused upon and discussed within the area of software engineering and usability. The combination of factors included in the framework means that is unique in solving a number of the problems that are found in these different areas within software engineering, especially in a rapidly changing marketplace. We also contribute to the field, through the qualitative element of the evaluation framework, which is inspired by ethnography and participatory design. It thereby makes a contribution that can improve practice in the field of software engineering, and that contributes to the theoretical work that is being performed within these different research areas.

Acknowledgements

Getting this far has been a long process. As a person who spent many years travelling, and exploring many avenues in life, life sometimes seems like a road movie, and this has definitely been a long and winding road. I would never have got this far without guidance and lots of help from, in particular, Dr. Kari Rönkkö and Professor Claes Wohlin. Thanks for helping me along, particularly when my maps ran out.

Thanks to all of the people at UIQ Technology AB, my industrial research partner, who have participated in my studies or otherwise helped my research. Particular thanks to Mats Hellman, Mårten Ahlberg and Mark Hinely.

I would also like to thank all the other colleagues at BTH who have helped me find my way along the road, including:

- My other colleagues in U-ODD: Olle Lindeberg, Jeanette Eriksson and Christina Hansson. Your comments and insights have been much appreciated
- Dr. Yvonne Dittrich and Professor Sara Eriksén for getting me started in research and opening up new avenues of exploration
- Jan Björkman for his enlightening discussions and nice encouraging cups of tea
- Dr. Annelie Ekelin for popping her head around the door, and asking how I was getting on.

Special thanks to my family, for their support, and help, and practical assistance when I have needed it, and for putting up with all those deadlines and late nights… and for not being *too* jealous when I have disappeared off to conferences in different parts of the world.

Thanks to Wordle (http://www.wordle.net) which is a toy for generating the "word clouds" that form the introductory page for each chapter in this thesis.

This work was partly funded by The Knowledge Foundation in Sweden under a research grant for the project "Blekinge – Engineering Software Qualities (BESQ)" (http://www.bth.se/besq)

Table of Contents


chapter
Software
one
engineering
quality
usability
testing
framework
Requirements
development
different
product
process
test
use
market
within
important
also
research
results
however
study
methods
tests
measures
attributes
challenge
success
look
failures
field
approach
marketplace
customers
agile
must
performed
organisational
rather
artifacts
measurements
measured
mass
subjective
much
testers
industrial
Cleanroom
mentioned
human
models
model
need
may
part
devices
becomes
complex
Driven
capturing
combination
usage
seen
knowledge
measuring
Together
number
challenges
processes
companies
Two
SUT
MDRE
thesis
users
focus
case
factors
UX
measurement
used
customer
future
discussion
basis
relation
role
measure
developing
situation
concerned
stakeholders
aspects
needs
features
improve
problems
negative
area
areas
activities
products
goals
understand
many
since
cases
developers
external
presented
Given
work
OPT
found
towards
provide
technology
even
whilst
good
common
changing
agility
design
Organisation
goal
time
find
system
user
new
context
behaviour
satisfaction
point
view
related
based
industry
information
way
experience
finding
objective
developed
see
mobile
metrics

Chapter One

1. Introduction

Software is all around us, in everything from aeroplanes and microwave ovens, to mobile phones. Some businesses, such as the mobile phone industry try to approach the broadest possible user category with one single product. To succeed in this places enormous demands on the quality of the product that is designed, produced and sold. Given the complexity involved making technology work under real world contingencies, i.e. the continuous and rapid evolution of society, the complex nature of software, understanding human nature and the social and cooperative aspects in the process of building large software applications [1], this poses a great challenge for both industry and research. As one way to approach the above challenges, the software engineering community has decided to focus on different quality aspects of software.

The principal focus of the work that is presented in this thesis is a usability evaluation framework for mass market mobile devices, allowing measurement, comparison and presentation of the usability of hand held devices. It measures usability empirically, on the basis of metrics that are complemented by a test leader's observations. By doing this it gives a demonstration of quality in use [2], from the customer and end-user point of view. So far, it is mainly a tool for measuring usability and quality in use, although the ambition is to adapt the tool in the future to capturing user experience to a greater degree than it already does. For a discussion of user experience, which is seen as a more encompassing term than usability, see Section 1.4 and Chapter 7.

The goal of the study that is presented in this thesis has been to develop and evaluate this evaluation framework under real world conditions and contingencies. The framework is intended to identify areas where product improvements must be made to improve usability and product quality. It should allow us to measure the usability of specific devices as well as compare different devices. The aim has been to develop a framework that is flexible and easy to use, both when developing new designs, and when evaluating implemented designs, to find possibilities for improvement; a framework that can be used to find product strengths and weaknesses, in order to direct development resources to the relevant areas, in order to improve the quality of the product. The results we have arrived at are of practical and theoretical interest within the industrial telecom sphere, particularly in the field of hand-held devices such as mobile phones, but also within the field of usability in general.

With its basis in the evaluation framework, and the way it makes use of quantitative measurements and qualitative judgments framed by a focus on usability testing, this thesis contributes to quality aspects related to: capturing adequate real world usage needs in a continuously changing society, and

supporting different information needs in the cooperative process of building the software.

The work we have done is related to three areas of software engineering, Market Driven Requirements Engineering (MDRE), Statistical Usage Testing (SUT), and Organisation and Product Together (OPT). The construction of the framework means that it contributes to solving a number of the problems that are found in these different areas within software engineering. The contribution that we make is presented in greater detail in the different sections of this chapter.

The study is presented as the results of two case studies. These cases studies are based on a series of tests, where the testing has been performed by using the evaluation framework concurrently in two countries, within two different organisations. This testing process was also a quality assurance of the test package itself, examining how efficiently and flexibly it worked in a complex industrial context, with complex relationships between customers, partners and end-users. These cases are presented in Chapters 5 and 6.

The first case, in Chapter 5, an inductive study, led to the conclusion that, in order to adequately address the challenge of usability testing, we must include the information needs of different stakeholders in the design and development process, and also address the challenge of adapting information to different time scales. This led to a discussion of different approaches to testing and presenting test results, where we found good reasons for what could otherwise be called "bad" testing, and where we placed different information needs on a scale ranging from agile to plan driven.

The second case, presented in Chapter 6, is a deductive study of different actors and their information needs, and is a follow-up to the work presented in Chapter 5. Here we have performed a survey where different stakeholders in the design and development process were asked to evaluate and prioritise different methods of presenting the results of usability testing.

In the remainder of this chapter, we look more closely at the background factors that have influenced this study and the contents of this thesis, in the area of mass market products, and the wider field of software engineering.

We begin by looking at why usability is so important for mass market products, where the challenge is developing general products that will appeal to as many customers as possible, in a market that is characterised by fierce competition. Following this, and in relation to this situation, we relate our work to the areas of MDRE, SUT, and OPT. Then we take a more detailed look at the concept of quality, and how it is related to usability work within software engineering, and in particular at the future challenges for quality work in the rapidly changing marketplace.

Following this is a brief discussion of the situation within software engineering, where there are problems implementing usability practices. These problems are grounded in a communications gap between the field of software engineering and the related fields where usability has been discussed and developed. The

section that follows this takes up the need that we have found for agile practices to be included in usability work within software engineering, where we find good organisational reasons for practices that may be deemed to be bad practice if we look at them from the prevailing view of testing. The chapter is concluded with a look at theories regarding software measurement, and how these are related to the metrics and measurement that are part of the evaluation framework.

1.1 Usability for mass market products

It is a considerable challenge to develop a usability evaluation framework for mass market products, where economic benefits are gained through approaching the broadest possible category with one single product. Whilst there is a need for targeting specific end-user groups [3], there is also an unwillingness to exclude other potential end-user categories [4]. When developing software for single organizations, end-user participation, empowerment and the development of adequate routines to be supported by technology are easier to identify, scope, and handle. In a mass market, such as the area of telecommunications, it is harder to identify and portray the complexity and representativeness of end-users [4]. Social and political aspects that influence the usefulness of products might even be filtered out by the evaluation techniques used in mass markets [5].

Particularly in regard to the development of hand-held devices such as mobile phones, the design and production of products is influenced by competitors launching new products and by technical magazines publishing reviews and comparisons of mobile features. Timing aspects give competitive advantages and influence design decisions. It is a branch that focuses on providing the market with new and improved technology rather than fulfilling end-user needs [6]. These branch characteristics challenge and bound the design space of usability tests.

The challenges mentioned above are common to the case studies presented in this thesis, and particular emphasis is placed on the fact that the results of performed usability tests must be adequately presented to different stakeholders. The information needs of the various stakeholders are dependent on different professional interests such as marketing, management or design.

In the light of the above, and in order to put the usability evaluation framework in relation to other movements within the area of software engineering that have connections to the work that we have performed, this chapter continues by relating our framework to Market Driven Requirements Engineering, Statistical Usage Testing, and Organisation and Product Together.

1.2 Related movements in software engineering

Work within the area of Market Driven Requirements Engineering (MDRE) has not focused specifically on the subject of usability in software engineering. Instead, MDRE has investigated and illustrated a number of different areas and has revealed complex market related contingencies that must to be managed in

software engineering if efforts are to be successful. Usability is only one of the identified challenges in this area. In this regard, this thesis provides the MDRE community with deepened knowledge concerning usability, whilst MDRE provides a context and a detailed map wherein we can place our results. We begin by providing an overview of MDRE, and how this thesis contributes with the addition of usability knowledge.

Market-driven software can be sold as Commercial Off-The-Shelf (COTS) products, or be combined with hardware in embedded systems [7]. All kinds of products, such as mobile phones, cars, aircraft, toys and games contain software, and market-driven software products are sold on an open market, where there is a vast range of potential customers. This means that diverse requirements have to be taken into account, and market-driven products are often developed in consecutive releases in a situation where there is strong competition [7].

MDRE focuses on products offered to many customers in an open market, in contrast to developing a bespoke product for one customer [8]. In MDRE, requirements are often invented for packaged "off-the-shelf" software offered to imagined customers in the mass market. This is a situation that differs greatly from the contract system, where requirements are elicited for a bespoke system for a specific customer [9]. Requirements are invented based on domain knowledge, business objectives, and a product vision [7]. There is no distinct set of users. Instead, there are potential users who are an imagined group who may fit into the profile of an intended user, and eliciting requirements from this group is one of the main differences between MDRE and customer-specific requirements engineering [7]. Early attempts to implement Personas [10] at the company, which was the starting point for our research cooperation with UIQ Technology, fell short precisely because requirements were invented based on domain knowledge, business objectives, and product vision. That is, the starting point was not based on user needs as is required in the Personas method.

Potts has characterised requirements elicitation as a communications process, taking place between the developer and the customer, but this cannot be seen as a simple communication process. One party does not elicit requirements from the other, and requirements are not relayed so that the other party can accept or reject them. This is particularly apparent for developers of off-the-shelf software, where requirements are produced through a collective and incremental design process [11]. In relation to the focus on usability that is central to this thesis, this is visible in Chapters 5 and 6 that deal with the differentiation of results depending on the professional role of the stakeholder who the results are presented to, and how this can affect the design and development process. It was also visible in the complex situation found when two organisations within the same concern performed testing in two different countries, where there were complex dependencies between the companies and different aspects of the products that are being tested.

In a market driven situation, there is major pressure on time-to-market, and it is important to release products according to a careful plan for requirements

prioritisation and releases [9]. Time to market is a survival attribute, and if the product is not released on time, there is a high risk of losing market shares [7]. Release planning, selecting an optimal subset of requirements for realisation in a certain release, is a major determinant of the success of a market-driven software product, since it is the point where requirements engineering for market-driven software development meets the market perspective. It determines the features and quality that are delivered to the customer, at what point in time [12]. Release planning is normally preceded by requirements prioritisation and resource estimation, and after this it may appear to be simple to select requirements and allocate resources until the resources are fully allocated. However, the selection of features and allocation of resources in MDRE has been found to be particularly complex [12].

There are several factors in the above that are related to the importance of the usability focus that is central to this thesis. By using everyday users in everyday tasks, the test captures aspects of the marketplace that are difficult to capture and formulate in a traditional RE process, and these factors are, as shown above, vital to the success of our product, which is obviously dependent on acceptance in the mass-marketplace. The evaluation framework is one way of deciding and quality assuring the subset of features that are important in the marketplace. The factors named above regarding the importance of time to market, and the need to adapt releases to the market situation, also provide an explanation for why it was found necessary to develop a usability evaluation framework that is lightweight, and in many ways can be regarded as agile.

Potts has found that one of the implications for the future of MDRE is that attributes such as time to market and usability matter more for off-the-shelf products than for bespoke systems, but that the requirements engineering community has concentrated more on factors such as requirements disambiguation and correctness [11]. When designers select features, they are balancing multiple objectives, not merely reflecting what the customer has said [11]. Potts concluded that an off-the-shelf system is only as good as the accuracy of the contextual assumptions made by the designers, and also found that it is important to evaluate likely system behaviour in concrete situations, since in these cases there is no customer who can provide definitive answers to questions [11]. In this study, the problem of finding the right balance amongst multiple objectives has led to the situation where UIQ Technology, whose customer could be as diverse as a mobile phone manufacturer, or an end user, was forced from the beginning to build up its own usability research competence. The people with this competence acted as proxies for the customers, in many cases for the end users whose opinions for many reasons difficult are to gather, and they could use this position to find the right balance between the multiple objectives that they were faced with. They also kept track of real end user needs and opinions about designs.

Carlshamre also states that there are difficulties attached to understanding and describing tasks, and that this is a major issue in the areas of human-computer interaction as well as requirements engineering. Although there are a number of task analysis techniques for capturing flows of activities with fairly static

structure, these techniques are less effective for describing dynamic and creative tasks [12]. These are the kinds of tasks that are common for the products that are the subject of MDRE. Once again, the user research performed within the company is vital, and qualitative observations play a key role.

Within MDRE for platform-based development of embedded systems, such as mobile phones, it is vital to find the right balance between quality aspects, also known as non-functional requirements ([8],[13]). Non-functional requirements are defined as requirements focusing on how well the software does something, as opposed to functional requirements, that focus on what the software does [14]. Due to competition in the marketplace, companies must understand and meet the quality needs of their customers. However, non-functional requirements (NFR) are still poorly understood, even though neglecting NFR is counted as one of the top ten risks of requirements engineering [14]. Paech and Kerkow conclude that the goal of future research within NFR should be aimed in four directions, in order to find a way of grasping the notion of quality, and to provide a common ground to talk about quality in different contexts, leading to an ability to manage the common ground in different projects. These directions are: to perform more empirical research into how NFR affect the performance of projects; to attempt to understand subjective vs. objective notions of quality; to apply ethnographical methods to gain an understanding of quality and negotiate quality and; to apply visualisation methods, to improve our ability to view quality, and assist humans in creative thinking [14].

In relation to the above concerning NFR, we have applied the ethnographic mindset to the study, together with knowledge from the area of participatory design. The evaluation framework we have developed is a form of requirements engineering, which has for example been used to validate the most important use cases for one of the customers/joint owners. It is specifically designed to capture user opinions regarding quality in use. This combination of validation and quality assessment on the basis of user preferences means that the framework can be used to discuss and improve the understanding of the nature of quality, and can promote discussions regarding the different aspects of quality. To further our quality work, we have also introduced new methods for visualising test results based on inspiration from the Kano model (see e.g. [15]), allowing us to show quality of use in a way that has a strong visual impact, also promoting discussions of the nature of quality. This focus, in line with the view of Paech and Kerkow, contributes to improving the knowledge of usability, and quality of use, within the area of NFR and MDRE.

Having looked at the area of MDRE, in order to set the evaluation framework in the context of the industrial area where we are working, and the special needs that attend it, we now take a further look further at software testing, More specifically, we look at Statistical Usage Testing, a technique in use within Cleanroom software development. Cleanroom software development is an approach to development that uses formal methods to support rigorous software inspection ([16], p. 532). According to Pfleeger and Atlee, the Cleanroom approach is based on two fundamental principles: to certify software with respect to specifications, rather than waiting for unit testing to find faults, and:

to produce zero-fault or near-zero-fault software ([17] p. 469). There are five key strategies of Cleanroom software development ([16], p. 532-533). These key strategies are: Formal specification, Incremental development, Structured programming, Static verification, and Statistical testing of the system.

Since we are interested in the area of software and system testing, and in particular testing based on the use of a device, we proceed by looking more closely at Statistical Usage Testing (SUT). The main objective of all testing is to validate that the system fulfils requirements, and although the focus is mostly on functional requirements, test cases can also address quality issues [18]. The two major approaches to testing are black-box testing, where testing takes an external view of the system and test cases are generated without knowledge of the internals of the system, and white-box testing, which takes an internal view of the software and aims for example to cover all of the paths in the code, or all of the lines of code [18]. Usage based testing is a black-box testing that is focused on detecting the faults that cause the most frequent failures, thereby maximising reliability. It requires a model of anticipated software use and quantification of expected use when the software is released [18]. We are not concerned with system internals, so our usability evaluation framework is based on the idea of black-box testing, starting as it does with e.g. scenarios prioritized by clients. What differs in our case is that executions of the chosen scenarios by a user are observed by a test leader. In this situation, users talk aloud at the same time as they act, thereby demonstrating and making visible their intentions and beliefs, giving a degree of transparency.

SUT is a form of usage based testing where the system is tested according to randomised test cases that are based on the probability of usage. The testing results are used to construct a quality model to determine mean time to failure, and other quality measures ([17] p. 470). In coverage testing, which is a form of white box testing, the aim can be to design tests that cover every path through the program. In SUT, however, a test developer draws tests at random from a set of all possible uses of the software, and the tests should be representative of the distribution of expected usage. This is claimed to be more cost effective than coverage testing at finding execution failures, defined as when the software does not do what it is required to do [19]. Since some execution failures will occur frequently, whilst others appear infrequently, it is also claimed that usage testing that matches the use profile of the software is more likely to find the failures that occur most frequently, whilst coverage testing is as likely to find a rare failure as it is to find a frequent failure. If the goal of testing is to maximise expected mean time to failure, usage testing, which concentrates on the failures that occur most often, must be an efficient strategy, [19].

The goal of the usability evaluation framework that we have developed is to reveal the extent to which a product can be used by specified users to achieve specified goals in a specified context of use. This is related to: the accuracy and completeness with which users achieve specified goals; resources expended in relation to the accuracy and completeness with which users achieve goals; freedom from discomfort, and positive attitudes towards the use of the product. The use cases chosen for testing may be (but are not necessarily) representative

of the use profile of the device or software, but the measurements are not concerned with failures of the software. Instead, they are concerned with individual attitudes towards the behaviour and appearance of the device as a whole. The presence and discovery of failures is of course an important facet of the testing, but even if failures do not occur, important results are reached by observing the user performing the use case. This juxtaposition of our goals compared to the goals of SUT makes visible the way in which we rely on the point of view of each person who has participated in the test, whilst SUT relies on a view external to the views of the actual users.

Despite the publicised success of Cleanroom techniques, with many reports in the literature suggesting that Cleanroom improves software quality, there remains criticism of Cleanroom, and of the use of SUT in certain circumstances ([17] p. 473). Pfleeger and Atlee point out that debate continues as to whether Cleanroom techniques are as superior as their supporters claim, and point out that it is of paramount importance to be sceptical of techniques that purport to solve major problems of quality and productivity, and of evaluations that attempt to convince us that one technique is superior to another ([17] p. 473).

The main focus within SUT has been on creating a usage model from a testing perspective (or as in our case the point of view of a person who has participated in the test, or the wishes of a customer), rather than creating a model from a requirements perspective, although approaches have been developed that attempt to remedy some of the shortcomings by e.g. combining SUT with use case modelling [18]. At first glance, SUT would appear to be a promising candidate for the kind of testing that we have been performing, but as we see, there are differences between the usability testing that we have been concerned with and the type of verification that is performed via SUT.

Apart from the factors mentioned above concerning the point of view of the users, the primary difference is that we are mostly concerned with examining people's attitudes towards the software system and hardware, rather than testing for failures. Another difference is that we are working in such a rapidly changing marketplace with a diversified customer profile, that it may be difficult to formulate and maintain a usage profile that will suffice to cover all of the expected use scenarios.

Since we are interested in both the human side of development, and how the data from testing is used in and affects the organisation, we now choose to look at an approach called Organisation and Process Together (OPT). OPT is an iterative improvement method, where steps include modelling the relationship between the organisation and the process, measuring properties of the relationship, and choosing organisational and process changes to be implemented [20]. In 1993, Seaman wrote that a basic shift was gaining support, moving focus from a product-centred view of development to a more holistic view including non-product characteristics. This shift was necessary to increase the quality of the software to the level expected by the marketplace. The article deals with the role of the organisational structure [20]. It is clear that the development process has a direct effect on the quality of the product, but the

structure of an organisation also has a significant effect on the ability to produce and maintain quality software. The OPT approach considers both of these elements together. The goal of OPT is to understand how well suited the development process and the organisation are to each other, and this entails asking how good a process is for an organisation and how good the organisation is for performing the process [20].

The focus on quality is a factor that is common to both our usability evaluation framework and OPT. However, OPT is as mentioned very much a process-centred approach to product improvement. Our usability evaluation framework, on the other hand is a much more product centred activity, even though organisation and process are important elements to be taken into account when studying how the usability evaluation framework is used and is useful in the organisation (see Chapters 5 and 6). The framework is not concerned with effecting changes in the relationship between the organisational structure and the development process. It is also a lightweight process, whereas OPT appears to be a complex methodology.

We summarise this section by looking at the challenges inherent in the areas presented above, and in which way the combination of factors in our framework contributes to the area of software engineering.

In relation to MDRE, we find that the framework solves a number of problems that are noted in the field. The framework works as a mechanism for managing the problematic communication between developers and customers. The customers in this case are end users of the product, and the framework thereby captures aspects of the marketplace that are found to be difficult to formulate, yet are vital to the success of the product. The agility of the framework, and the speed with which results are reached, is also an important factor, since success in the marketplace is found to be dependent on adapting quickly to the rapidly changing demands of the customers. The fact that the test leaders are usability experts, and act as proxies for the end users when presenting results, also means that the assumptions that they make are grounded in the concrete situation in the marketplace, and are not contextual assumptions made by designers. The construction of the framework, with its basis in ethnography and participatory design, and its presentation methods, is well in line with the goals of future research within non-functional requirements, needed to give a common ground for grasping and communicating about quality in different contexts.

Compared to SUT, we find that our testing strategy, whilst being a black-box strategy in the same way as SUT, still gives a degree of transparency where the users reveal their intentions and beliefs when performing the testing, and the test leader can capture this information. The purpose of our testing is different to the purpose of SUT. SUT is concerned with finding usage patterns that result in failures, but our testing is more concerned with attitudes towards the behaviour and appearance of the device as a whole, and important results are reached even if failures do not occur, since we observe the user who performs the use case.

In relation to OPT, we find a common factor in that our framework and OPT are both based on an ambition to include the human side of the development, but that our framework is much more product oriented, rather than being process oriented. It is also an agile process, rather than a complex methodology. In the market situation where the product is being developed, we find that this agility and product focus are important factors for success in the type of rapidly changing context and marketplace where many companies are active.

In summary, the combination of factors above mean that the evaluation framework solves a number of the problems that are found in these different areas within software engineering, and thereby contributes to improving improve practice in the field of software engineering, and to the theoretical work that is being performed within these different research areas.

1.3 Quality – a vehicle for usability work in software engineering

The research for this thesis has been conducted as part of the BESQ research environment that was launched in July 2002. The Knowledge Foundation (KK-Stiftelsen) was the main initial sponsor of the initiative together with matching funds from industrial partners. The initiative has initially run as a research programme with the objective to create a strong research environment within software engineering. As one result the BESQ (Blekinge Engineering Software Qualities) Research Centre was established 2007. The mission in BESQ RC is to provide support for quality balanced software (QBS) to its partners. Individual projects, most often in close industrial collaboration, should contribute to the overall mission [21].

This thesis is a contribution to quality balanced software since the main task of the usability evaluation framework that has been developed is to validate that the products under development satisfy user needs and provide value to the users.

Osterweil [22] states that product quality is becoming the dominant success criterion in the software industry, and believes 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 Osterweil states 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 [23]) and even to changes in legislation that make executives responsible for damages caused by faulty software.

One prevailing 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 [24].

1.4 Quality – future challenges for usability professionals

A large part of the material in this chapter has dealt with the areas of usability and quality, and how our study and the evaluation framework are related to these. Previously in the chapter, however, we mentioned the fact that the goal has been to develop the evaluation framework so that it becomes a tool for measuring user experience (UX). When we introduced the term, no more detailed definition was given beyond the fact that UX is seen as a more encompassing term than usability. In this section we look more closely at UX, and how it is related to the way that usability and quality have been defined and measured.

When comparing usability and UX, usability today can characterised as being based on a pragmatic, objective, and negative approach, whilst UX is characterised as being holistic, subjective and positive. It is these three aspects that differentiate usability from UX [25].

Usability has been concerned mainly with the pragmatic side of judgement, with a strong focus on the users and their tasks, and how they accomplish these tasks. Usability is equated with task related quality, and is based on such concepts as are defined in e.g. the different ISO standards, where usability and quality of use are concerned with efficiency, effectiveness and satisfaction [26]. Usability takes an objective view of quality, basing its judgements of usability on observations of product use, rather than on opinions (even though registering user satisfaction can be said to be one approach to capturing the subjective aspects of usability). By concentrating on barriers, frustration and failures, usability testing (in much the same way as other software testing) has concentrated on the negative, and identifying and removing the negative is and will remain an important part of product design and verification.

With its focus on these areas, and its basis in the ISO standards, and influenced by areas such as cognitive psychology, work psychology and human factors, usability testing, via observation and surveys, has become successful at identifying “hygiene factors” ([27]). These are dissatisfiers, whose absence may lead to dissatisfaction but whose presence does not necessarily lead to satisfaction. These factors can also be equated with the “must be” factors in the Kano model, which we return to later in the thesis [15].

The field of usability has long experience of working in this fashion, and through this has come a long way towards satisfying industrial needs. Discussions of this area are however lacking within software engineering, and this thesis is therefore an important contribution in this area.

If we look towards the future, a major challenge is working with UX. Much work is being done within the field, both theoretical and industrial, but much remains to be done, and in many ways it is still uncertain how UX differs from the traditional usability perspective [28]. If UX is to be accepted as a topic in its own right, it must differentiate itself from and add to the traditional view of interactive product quality [25].

Where usability is seen as concentrating on the pragmatic, objective, and negative, UX is holistic, subjective and positive [25]. An important term in UX is "hedonic", meaning related to, or characterised by, pleasure. UX takes a holistic approach, taking into account and finding a balance between both pragmatic aspects and hedonic aspects – non-task related aspects of product use and possession, such as beauty, challenge and joy. In this way, UX is subjective, since it is interested in the way people experience the products they use. Both usability and UX can concentrate on efficiency, effectiveness and satisfaction, but UX attempts to understand satisfaction in terms of fun, pride, pleasures and joy, and attempts to understand, define and quantify these elements [29]. These hedonic aspects are important, firstly because they colour the way that individuals experience owning and using a product, thereby affecting future behaviour, and secondly because they affect how people pass on to others their experience of owning a product. This is related to what sells well in the marketplace, and becomes more and more important in the type of market driven area where this study has taken place. The positive types of product experience connected with the hedonic features mentioned above are therefore important in UX. The focus given to the negative that is found in the usability view of product quality is not regarded as unimportant, but what is emphasised is that the positive is not simply the absence of the negative.

The positive is a further dimension of quality, and high levels of satisfaction result from the presence of hedonic qualities, rather than the absence of displeasers. Once again, this can be compared to the Kano model, where the presence of attractive requirements can sometimes be found to outweigh a lack of "must have" features [15]. It may be found that usability has been good at measuring the "must have" attributes that a customer expects, and even the qualities that are explicitly demanded by the customers, whereas UX is better at finding the "attractive attributes" that are unexpected and delight the customers by their presence.

In a situation where subjective factors such as beauty, fun, pleasure and pride, are important, where a product can be seen as an expression of your personality or identity, UX says that capturing these factors becomes more important than being able to objectively measure usability. It becomes important to focus upon motivators and satisfiers. For mobile phones, this movement towards hedonic qualities is in many ways driven by advances in technology, enabling better graphics on larger screens, with better algorithms and processors, and longer battery life. The preconditions improve constantly, whilst users demand more and more as they get used to the improvements that are made. As a result of this, usability work in the future will see a move away from the pragmatic, objective and negative views of product quality, and quality in use, towards holistic, subjective and positive views. This movement is already apparent, at least in the field of mobile phones, as mentioned here. In our framework, the presence of the usability expert who has the role of a test leader, observing the testing and interacting with the testers, is one step towards capturing and measuring the subjective factors that make up satisfaction.

The ever shifting situation leads to challenges within the areas of both usability and software engineering. The dynamics of the marketplace, and constant advances in technology, where constant change is an expected factor, mean that agility becomes more important. We return to this discussion of agility in Chapter 5. Since UX is methodically incoherent and no accepted definition of UX yet exists [29], and there is a low level of maturity within the field of usability concerning UX, it is natural that this thesis concentrates mostly on usability within the field of software engineering rather than on UX.

1.5 Underuse of usability methods or a lack of publications?

During the past two decades, the human-computer interaction (HCI) community has developed many promising user-centred methods. However, many of these methods are still underused by software developers and organizations who find them difficult to implement. One suggested explanation is that the methods are developed independently from the software engineering community. There have been a number of conferences workshops and journal special issues dealing with the theme of bridging this regrettable gap between software engineering and HCI (see e.g. [30-35]). However, empirical studies of software engineering have demonstrated in many cases that successful in-house usability practices exist in software engineering organizations although they are not published, either at scientific conferences or as organizational documents [36]. The latter claim, that successful practice exists, contradicts the first claim, that usability methods are underused in software engineering.

A related issue is, as discussed previously, that usability is a moving object, and therefore the current meaning of it in the research discourse might not correspond to established quality models in today's software engineering industry. Another issue is that the current state of empirical studies is criticised for not building on previous research results, particularly those achieved outside the researchers' own domain [37].

In the light of the above, individuals with some knowledge of usability might find our implementations to be fairly standard usability. This is true in the sense that we have taken a decision to heavily build upon the research results of others, and the fact that software engineering lacks publications and academic discourses in this subject, but there are features of our approach that differentiate it, and we discuss this briefly below. We also present a fresh picture of usability work in a contemporary industrial context, and base our work on a broad spectrum of research areas.

For the reasons given above, it is important that we clarify what our own contribution consists of. There are several features and aspects that are new and challenging in the latter versions of the evaluation framework that have been the subject of this study. These are: the particular mix of qualitative and quantitative usability methods in the test package, ensuring that it is able to target a number of different stakeholders; experience in applying these methods in the technology-focused and rapidly changing area of mobile phones; the challenge of addressing end users in a mass market; the challenge of mobility;

the challenge of finding presentation models for the set of stakeholders that the evaluation framework has to address; the developmental adaptation, i.e. making sure that the usability evaluation framework can be used in different stages in the industrial software development process – this adaptation includes ensuring the agility of the framework, something that will be discussed in more detail in Chapter 5.

1.6 Demands for an agile approach to testing

The demand for continuous and rapid results in a world of continuously changing business decisions often makes the approach to testing mentioned above, (relying on complete, testable and consistent requirements, traceability to design, code and test cases, and heavyweight documentation), impractical or impossible, and this points to a need for agility. At a keynote speech at the 5th Workshop on Software Quality, held at ICSE 2007 [38], 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 kind of situation demands an agile approach [24]. 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.

If quality is 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 [23].

Within software engineering, there are many types of testing, in many process models, (e.g. the Waterfall model [39], Boehm's Spiral model [40]). Testing has been seen as phase based, and the typical stages of testing (see e.g. [16], [17]) when developing large systems are *Unit testing, Integration testing, Function testing, Performance testing, Acceptance testing*, and *Installation testing*. 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 [17].

Agile software development has changed how software development organisations work, especially regarding testing [41]. In agile development, exemplified here by Extreme Programming (XP) [42], testing is performed continuously by developers, the tests should be isolated, i.e. should not interact with the other tests that are written, and should preferably be automatic, even though a recent study of testing practice in a small organisation has shown that not all companies applying XP automate all tests [43]. Tests come from programmers and customers, who create tests that serve, through continuous testing, to increase their confidence in the operation of the program. Some XP

teams 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 [42].

The role of the tester is a matter of debate. In both of the above cases 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. Bret Pettichord [44] for a discussion regarding this). Pettichord, a test automation engineer, 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.

The thesis presents an approach to achieving quality that is 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. We demonstrate how an agile approach is adopted in the usability test. Within usability and user experience testing it becomes apparent that the test leader needs to possess traits other than those demanded in the engineering and technical areas. Here, the focus is not directed towards understanding and discovering faults in software; it is instead directed towards understanding human beings, and how to best treat them to get them to reveal their attitudes and feelings towards and understanding of our products – things that they themselves may sometimes be unaware of – and even discover why they feel and act the way they do.

1.7 Measurement in software engineering and in our study

Software measurement is a collection of fringe topics, generally referred to as software metrics, which have not been regarded as mainstream software engineering ([45] p. 20). Metrics can be described as "the discipline of counting things and observing and profiting from patterns found among the things we count" ([46] p. 15), and there are at least three different reasons for collecting metrics: to discover facts about the world; to steer actions, and; to modify human behaviour (ibid. p. 17). Since software development cannot be equated with production, software measurement is not simply a case of listing standard measures, and measures must be chosen, adapted and used according to goals of interest, and the usefulness of software measures must be judged in context. Measurement activities, both academic and practical, have moved

towards measurement methodologies, concerns regarding introducing measurements into real environments, and feedback activities to enable learning within and between projects ([47] p. 660). Metrics programs are a vital part of every serious attempt to improve software processes, but despite a number of success stories, most companies appear to experience difficulty in implementing metrics programs as part of their software process improvement activities. This is probably a result of the complexity and uncertainty involved in implementing metrics activities ([48] p. 287).

A software measure is a way of mapping from a set of objects in the software engineering world, such as projects, processes and products, to a set of mathematical objects, such as numbers, or vectors of numbers. A software measure can be used to characterise a property of software engineering objects quantitatively ([47] p. 646). Software measurement is a method of applying software measures to software engineering objects to achieve a predefined goal. These goals vary according to which objects are being measured, the purpose of the measurement, who is interested in the measurements, which properties are measured, and the environment in which they are measured. Measurement must be sensitive to different goals, since each goal requires a specific set of measures, and the data collected requires different interpretations ([47] p. 647).

Fenton and Pfleeger state that there are three general classes of entities in software that we wish to measure. These are: processes, which are collections of software-related activities; products, which are artefacts, deliveries or documents resulting from processes, and; resources, which are the entities required by process activities. Each class of entities has internal and external attributes. The internal attributes can be measured solely in terms of the process, product or resource itself. In other words, they can be measured by studying the entity in isolation from its behaviour. External attributes can only be measured on the basis of the behaviour of the product, and how it performs in relation to its environment ([45] p. 74). Managers and users are often interested in external attributes. A user is for example directly affected by the behaviour of a system, and may be interested in reliability, usability and portability. External attributes are however usually harder to measure than internal attributes, and can in many cases only be measured late in the development process ([45] p. 75). Measurable external product attributes include usability, integrity, efficiency, testability, etc., and since external product attributes depend on both product behaviour and environment, measures must take both of these characteristics into account ([45] p. 78). When measuring external product attributes, it is important to define quality in terms of specific attributes of interest to the user, and we must know how to measure the level of these attributes in the product. This allows us to specify the quality attributes in a measurable form ([45] p. 337).

Measurement activities must be motivated by specific goals that must be clearly defined, and easy to understand, and the objectives of the measurement must be tied to what managers, developers and users need to know ([45] p. 12). Measurement is important to help us understand what is happening, to control what is happening, and to improve processes and products, but no matter how

the measurements are used, it is vital to manage the expectations of the people who are going to make decisions on the basis of the measurements ([45] p. 13).

The term software metrics covers many activities, most of which involve software measurement in some way. These include: cost and effort estimation; productivity measures and models; data collection; quality models and measures; reliability measures; performance evaluation and models; structural and complexity metrics; capability-maturity assessment; management by metrics, and; evaluation of methods by tools ([45] p. 14). The area that is most closely connected to the work presented in this thesis is within quality models and measures. Quality models as described by Fenton and Pfleeger are usually constructed in a tree-like fashion, where upper branches hold high-level quality factors of software products that we would like to quantify, such as usability and reliability, efficiency etc. These quality factors are composed of low level criteria, such as communicativeness, accurateness, consistency etc., which are easier to understand and measure than the factors themselves, and where metrics are proposed for the criteria ([45] p. 17).

To briefly summarise the above, our usability evaluation framework for testing mass market devices is a contribution to the field of software engineering, as far as developing products with the features and quality that are required to succeed in the competitive marketplace. The framework is designed and developed on the basis of theories and practice in the area of usability, and is flexible and easy to use, both when developing new designs, and when evaluating implemented designs, to find possibilities for improvement. However, it extends these theories, and the lessons learned and presented in this thesis can be used to instruct practice.

The combination of factors that are found in the framework means that it solves a number of the problems that are found in several different areas within software engineering, and the results we have arrived at are of practical and theoretical interest within the software engineering community and the industrial sphere.

The framework is a tool for measuring usability, but given the presence of a usability expert who has the role of a test leader, observing the testing and interacting with the testers, it also a step towards capturing and measuring the still vague concept of user experience, which is still very much an emerging research area.

Our framework fills a gap, with empirical knowledge of the needs of different stakeholders, and about where different elements belong in the software engineering development processes. Our work includes a discussion of the need for agility, which has not so far been focused upon and discussed within the area of software engineering and usability. We also contribute to the field, through the qualitative element of the evaluation framework, which is inspired by ethnography and participatory design.

This concludes the introduction to our field and the theoretical and practical background to the work that we have done.

Chapter 2 presents the methodological approach that has been used in the study, with our combination of action research and Cooperative Method Development, and our case study approach, combining ethnography, participatory design and grounded theory. It also gives a brief introduction to the case studies that are the basis of the thesis, and the methodology used in each case study.

Chapter 3 presents the areas of usability, usability testing and usability metrics. For reasons given previously we have looked to fields outside of mainstream software engineering, and have chosen some of the standard works on usability, more common in the field of HCI, and the standards accepted by industry. Together, these provide a background to the usability evaluation framework, and allow comparison of the structure of and use of the evaluation framework with what is written about the field.

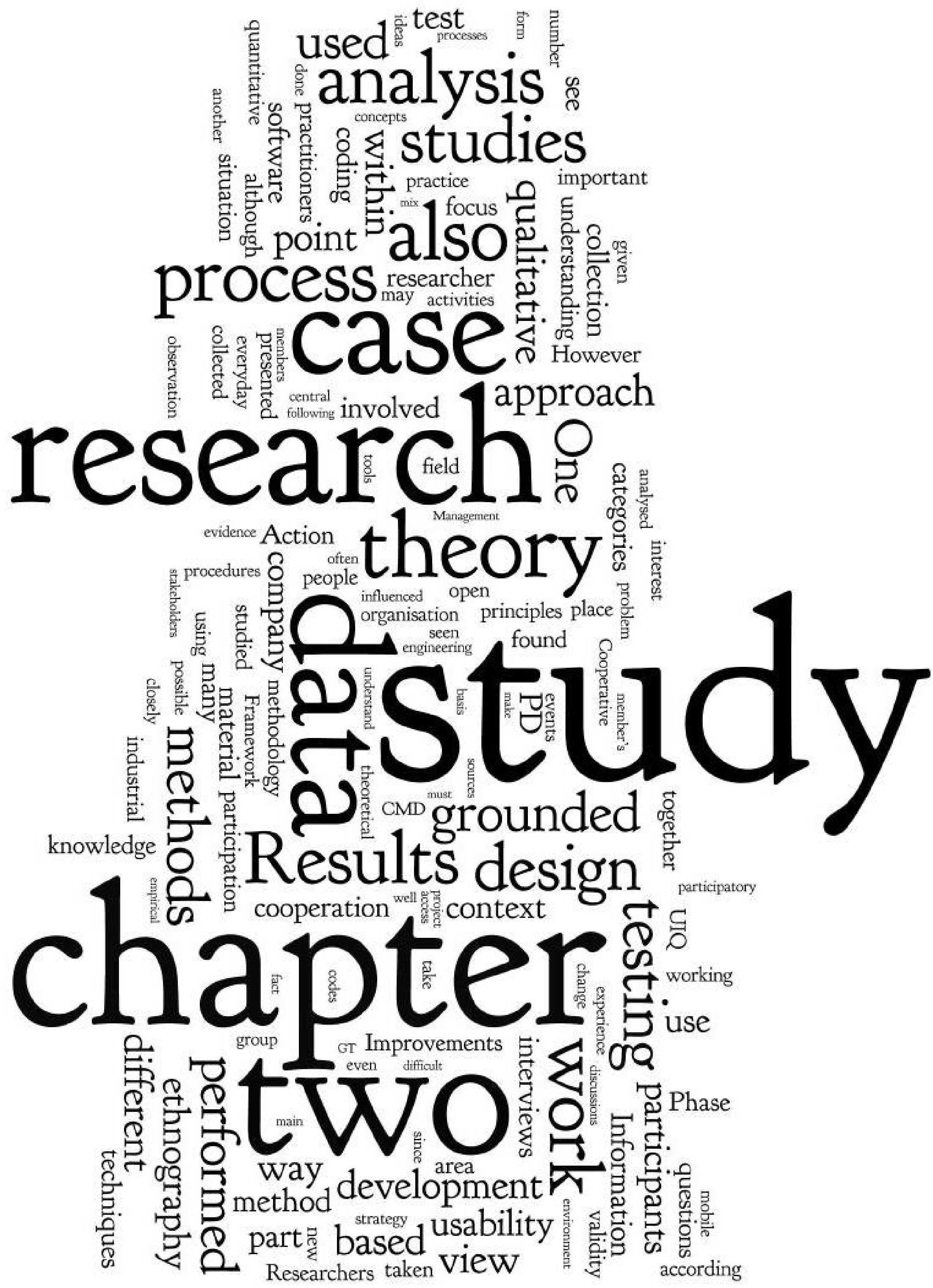
Chapter 4 is a description and explanation of the evaluation framework itself, including the philosophy that the framework is grounded in, the mechanics of the test and its presentation, and how we as researchers have affected the design and use of the test.

Chapter 5, which deals with agile and plan-driven processes, presents the first case study, where different stakeholders in the design and development process were found to have different information needs, and to need information on different time scales.

Chapter 6 is a study of different actors and their information needs, and is a follow-up to the work presented in Chapter 5. It is based on a small survey where different stakeholders in the design and development process were asked to evaluate and prioritise different methods of presenting the results of usability testing.

Chapter 7 is a brief look at usability and how it is related to the emerging field of user experience, which is an area that is of growing importance within the field where we are active.

Chapter 8 contains a discussion of the findings made in this thesis, conclusions that we have reached, and some directions for future work, based on our findings, and on the state of the industry and the research field today.


test
ideas
processes
form
number
used
analysis
see
quantitative
done
another
concepts
studies
software
practitioners
coding
within
situation
although
practice
important
mix
focus
qualitative
understanding
collection
given
point
also
process
researcher
may
activities
case
However
observation
collected
everyday
presented
members
central
following
involved
approach
research
One
tools
field
categories
analysed
Management
evidence
Action
theory
interest
often
procedures
company
people
influenced
open
stakeholders
using
studied
organisation
principles
place
problem
seen
engineering
found
Cooperative
possible
closely
many
methodology
understand
basis
make
PD
events
member's
data
study
industrial
material
Framework
theoretical
sources
must
CMD
grounded
together
methods
knowledge
participation
Results
design
participatory
empirical
cooperation
well
access
project
context
testing
UIQ
chapter
fact
codes
take
change
experience
working
use
group
GT
Improvements
even
difficult
different
ethnography
performed
two
interviews
work
discussions
main
since
Phase
participants
Informations
way
area
development
method
strategy
usability
questions
mobile
techniques
part
new
based
view
environment
validity
Researchers
taken
according

Chapter Two

2. The research cooperation & methodology

This chapter gives a description of the background to and the history of the research cooperation, the research group and environment, and the company that participated in the research cooperation. It also presents the research methodology used in the study, which has been based upon the research and method development methodology called Cooperative Method Development (see e.g. [49]).

The research has been performed as a series of limited case studies, where the fieldwork has been influenced by ethnography and the principles of participatory design, and the research material has been analysed in an approach that has been influenced by grounded theory.

All of these approaches are presented and set in relation to the work that has been performed in this study. The specific approach that has been used in the case studies is presented. The actual case studies are presented as chapters in the thesis. The chapter is concluded with a brief summary.

2.1 The research group and research environment.

The research group that has supported, framed and influenced this study is U-ODD (Use-Oriented Design and Development) [50], at Blekinge Institute of Technology (BTH), which is part of the research environment BESQ [51]. U-ODD takes on the challenge of approaching software engineering via use-orientation, influenced by the application of a social science qualitative research methodology and the end-user's perspective. The human role in software development has been found to be an area needing further research in software engineering. The task of understanding human behaviour is complex and necessitates the need for using qualitative methods, since quantitative and statistical methods have been found to be insufficient for this task [52]. U-ODD work to understand human behaviour, in a situation of cooperative method development together with industry, by developing knowledge in and applying qualitative research methods taken from the social sciences. The strength of qualitative methodologies is in exploring and illuminating the everyday practices of software engineers, through observing, interpreting and implementing the methods and processes of the practitioners. Researchers and practitioners cooperate in an iterative process to design solutions based on the perspective of those who work on the “shop-floor”.

2.2 The industrial partner and their product

The industrial partner has been UIQ Technology AB, which was established in 1999 and closed down in January 2009, due to changes in the structure of the industry, where UIQ Technology’s product was released as open source. This meant that the company was left without a paying customer base. In the spring of 2008, the company had more than 320 employees in Sweden, and around 400

employees in total. The company was situated in the Soft Center Science Research Park, Ronneby, Sweden and was a wholly owned subsidiary of Sony Ericsson and Motorola.

One of the company goals was the creation of a world leading user interface for mobile phones. Their focus was "to pave the way for the successful creation of user-friendly, diverse and cost-efficient mobile phones" [53]. They developed and licensed an open software platform to leading mobile phone manufacturers and supported licensees in the drive towards developing a mass market for open mobile phones.

UIQ Technology AB focused on usability from the beginning, and their ambitions within the area of usability were demonstrated by the fact that they employed a number of usability experts. With regard to usability and user experience, requirements handling at the company was the responsibility of Product Planning, whilst Software Development was responsible for the breaking down of requirements and evaluations of feasibility. Within Software Development the first breakdowns were done by a System Design team. Then, work was taken over by teams that are responsible for each component. The usability experts were part of System Design, and they performed validation of the proposed designs. They were also consulted by Product planning on elements of concept design. In general the usability experts entered the process somewhat too late in the process, and were regarded as simply one input among many others for deciding the requirements. The fact that end user input is important was a conviction shared by UIQ's clients, although they did not necessarily share the same methodological standpoint; indeed they often had different perspectives and different priorities and ways of representing users.

The product, UIQ (see Figure 2.1), is an open, media-rich, flexible and customizable software platform. It is pre-integrated and tested with Symbian OS, which is an industry standard operating system for mobile phones, licensed to leading mobile phone manufacturers. It provides core technologies and services. Above Symbian OS in the structure is the Development Platform, consisting of the Application Framework, System services, the GUI toolkit, and a Java solution allowing Java applications to run. Above that is the Application suite, which includes key functionality for a mobile phone, including communication, Personal Information Management (PIM), Information, and Utilities, such as the calculator, viewers, and remote synchronisation. The uppermost layer is the user interface [54].

Having looked at the actors involved in the research, and the product that is the subject of the research, we now proceed to look more closely at how the cooperation took place. The research methodology used in the work leading to this thesis is based on a case study and grounded theory approach, inspired by participatory design and ethnography. In the following section, these different methodologies and approaches are presented and discussed in relation to the work that has been performed.

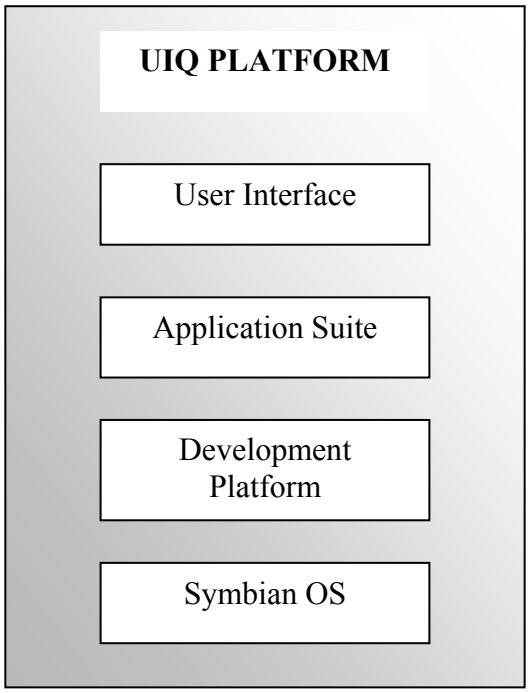


Figure 2.1. The UIQ platform

2.3 Action research and Cooperative Method Development

The process of cooperation is action research according to the research and method development methodology called Cooperative Method Development (CMD) (see [49] [55], [56] and ([57], Chapter 8) for further details). Action research is a "vague concept but has been defined as research that involves practical problem solving which has theoretical relevance" ([58] p. 12). It involves not only gaining an understanding of a problem, and the generating and spreading of practical improvement ideas, but also the application of the ideas in a real world situation and the spreading of theoretical conclusions within academia [58]. The purpose of action research is to influence or change some aspect of whatever the research has as its focus, and improvement and involvement are central to it ([59] p. 215). A central aspect of action research is collaboration between researchers and those who are the focus of the research, and their participation in the process, and the terms *participatory research* and *participatory action research* are sometimes used as synonyms for action research ([59] p. 216). Action research often takes the form of a spiral process, involving observation, planning a change, acting and observing what happens following the change, reflecting on the processes and consequences, and then planning further action and repeating the cycle ([59] p. 217).

Cooperative Method Development (CMD) is an approach to research that is a combination of qualitative social science fieldwork, with problem-oriented method, technique and process improvement [49]. CMD has as its starting point existing practice in industrial settings, and although it is motivated by an interest in use-oriented design and development of software, it is not specific for

these methods, tools and processes. In CMD, the action research process consists of the following three phases (see Figure 2.2, from [60]):

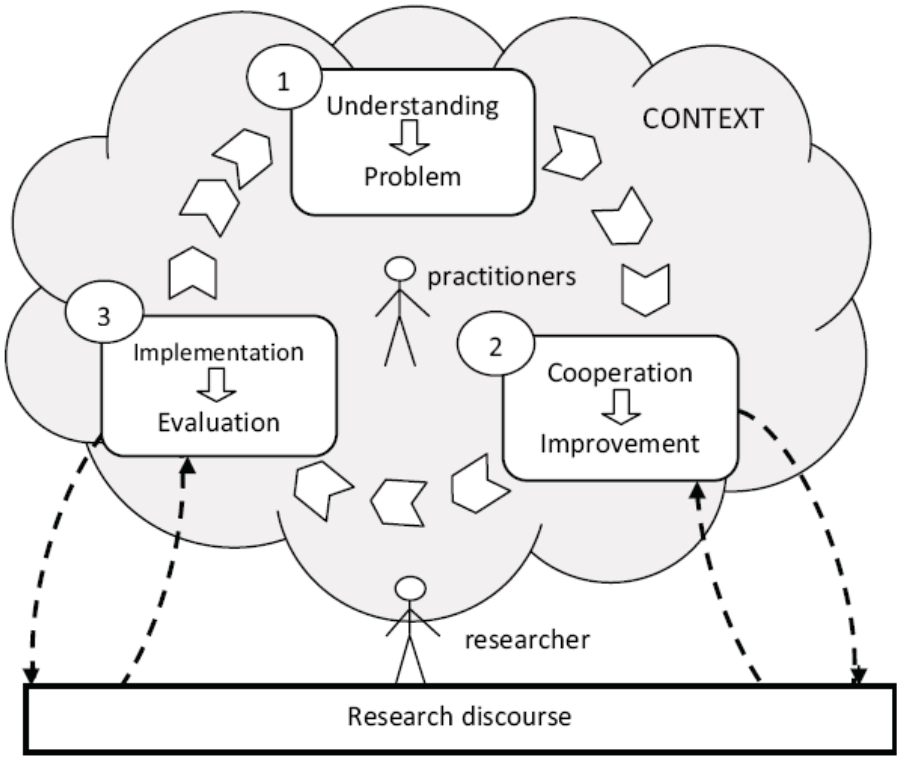


Figure 2.2. Cooperative Method Development

Phase 1 – Understanding Practice. The research begins with empirical investigations aimed at understanding and explaining existing practices and designs from a practitioner's point of view, based on their historical and situational context, to identify aspects that are problematic from the practitioner's point of view.

Phase 2 – Deliberate Improvements. Results from the first phase are used in a cooperative fashion by the researchers together with the practitioners involved, as an input for the design of possible improvements. The outcome of this phase is the deliberation of measures that address some of the identified problems and that are expected to improve the situation at hand.

Phase 3 – Implement and Observe Improvements. Improvements are implemented, and the researchers follow these improvements as participant observers. The results are evaluated together with the practitioners. In this evaluation, the concrete results are summarised for the companies involved, and they build a base for the researchers to evaluate the proposed improvements.

When comparing the phases contained in action research with the phases in CMD, the way that CMD is influenced by ethnography becomes evident in phase 1, Understanding practice, where the members' point of view is emphasised. In phase 2, Deliberate improvements, the CMD approach

emphasises the cooperative aspect. In phase 3, Implement and Observe improvements, two separate phases in action research are condensed into one phase, where we also, as software engineers, emphasise the aspect of evaluation.

Apart from the three phase approach presented above, CMD is built upon a number of guidelines, which are:

- Ethnomethodological and ethnographically inspired empirical research, combined with other methods if suitable
- Focusing on shop floor software development practices
- Taking the practitioners' perspective when evaluating the empirical research and deliberating improvements
- Involving the practitioners in the improvements

The intention of CMD is that is should be a framework that can be further detailed in relation to specific projects, and can even be limited to the first phase in the model if no improvement is decided upon ([49] p. 233). The research performed in this study has been formed in accordance with the principles that are the foundation of this approach and has followed these three steps. The following is an illustration of one of the CMD cycles applied in this study.

The first phase, understanding practice, began in 2005 when the researcher was introduced to the company, and began his studies. He was given a workplace at the company, and spent the first part of his time becoming acquainted with the company and its organisations whilst doing background studies of the area of interest. During this period, he interviewed members of staff, observed working procedures, and studied the development methodologies and project models in use at the company. At the same time he performed literature studies in order to become acquainted with the theoretical background to the field.

The second phase, deliberate improvements, was based on the work performed in phase one. As a concrete example of this, the researcher had identified as an area of interest methods for presenting the results of usability testing, and had studied the theory and use of the Kano model [15]. Based on this, he distributed research articles to the practitioners at the company, and organised discussion and seminars to examine how this could be of use in the usability evaluation framework that was being developed. As a result of the discussions and seminars, a new way of presenting the results of the usability testing was formulated.

In phase three, implement and observe improvements, the researcher observed how the changes in presentation had taken place, took part in meetings where the results were presented to and discussed with senior managers, and discussed these changes with the practitioners, to gain an understanding of what the improvements meant in the everyday work of testing and presenting the results of usability testing.

This cycle, as did the other cycles in the study, also conformed to the principles given above, by using a combination of ethnographically inspired fieldwork,

combined with document studies, in order to grasp and understand the practices of the workers who were involved in the everyday work of testing and designing the evaluation framework. The work was performed together with the practitioners, and the improvements were designed, tested and evaluated together with these practitioners.

In this particular application of CMD given above, the work is based on a case study [61] and grounded theory [62] approach. In what follows, we will look more closely at the case study and grounded theory approaches, and at ethnography and participatory design, and relate the theories to the work that has been performed in our study.

2.4 Case study

According to Yin, a case study is "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" ([61], s. 13). It is a well-established strategy, where the focus is on a particular case, taking the context into account, and which involves multiple methods of data collection; the data can be both qualitative and quantitative, although qualitative data are almost always collected ([59] s. 178). The need for case studies has its basis in the desire to understand complex social phenomena, and it is a suitable strategy when "how" or "why" questions are being asked, and where the researcher has little control over events ([61], p. 7). A case study approach allows the retention of characteristics of real life events, including organisational and managerial processes [61].

There is a wide interpretation of what the case can consist of, and it can range from studies of individuals, a set of individuals, a community, a social group, an organisation or institution, or a study of events, roles and relationships ([59]). Case studies can be explanatory, exploratory or descriptive, and the choice of strategy depends on the type of research questions posed, the degree of control the investigator has over events, and the degree of focus on contemporary or historical events ([61], p. 5).

Some important points regarding a case study ([59], p. 179) are that it is:

- A *strategy*, meaning an approach, rather than a method
- Concerned with *research* in a broad sense, including e.g. evaluation
- *Empirical*, in that it relies on the collection of evidence about what is going on in a setting
- It is a study of the *particulars* of a specific case, and an important concern is how the particular case can lead to generalisable results, and how this can be done
- The focus is on a *phenomenon in context*, and that
- It utilises *multiple methods* of evidence or data collection

A case study is an example of what Robson [59] calls "flexible design" research. A flexible design evolves during data collection, and data are often

non-numerical, so studies of this kind are often referred to as qualitative research. However, Yin clarifies that case studies can be based on any mix of quantitative and qualitative evidence – they can even be limited to quantitative evidence ([61], p. 14).

The case studies that are presented in this thesis are studies of an organisation, although focus has been placed on the individuals that are part of this organisation. Although it has concentrated on the role of the individuals, it has also looked at the organisational and managerial processes that steer the work that is performed. The study has been a combination of exploratory and explanatory, exploring the nature of usability testing in this particular industrial context, and attempting to explain why the testing is performed in the way that it is, given the factors that bound and steer it.

In our study, we have examined the theory and practice of usability testing in a concrete situation within the context of industrial design and development. We have studied how the testing is performed, to discover why it is performed in this way, in order to understand and to find ways of improving it. Since we have been working according to an action research strategy, we have been able to influence events to some degree, but the main influence in this context has been from the marketplace, and from the organisation, with its complex mix of industrial dependencies. Thus, there are many external factors that affect the work and its performance.

We have collected many different types of data from our observations and document studies, and although much of the data has been qualitative, we have also collected quantitative data, in much the same way as the work that we have studied has also consisted of the collection of qualitative and quantitative data.

There are a number of criticisms that have been levelled at the case study strategy ([61], p. 10-11). The greatest concern has been a lack of rigour, and in many cases this is a result of the researcher not following systematic procedures or allowing biased views to affect the direction of the study. A second criticism is that a case study, based on one case, cannot provide a basis for generalisation. A third complaint is that case studies take too long and result in massive, unreadable documents.

In order to refute some of the criticisms above, it is important to plan and design the research methodically and carefully. The characteristics of a good flexible design are ([59], p. 166):

- Rigorous data collection, where multiple data collection techniques are used
- The study is designed with the assumptions and characteristics of a qualitative approach to research
- The study is informed by an existing strategy, where the researcher identifies, studies and employs one or more traditions of enquiry. The tradition need not be pure, and procedures can be brought together from several traditions.

- The study begins with a problem or idea that the researcher seeks to understand.
- The researcher uses detailed methods for the collection, analysis and writing
- Analysis is performed on several layers
- The report is written clearly and in an engaging fashion, allowing the reader to experience "being there"

This research has been performed as part of a research group, and within in an environment, that is characterised by a tradition and the assumptions of qualitative research. The problem we have been seeking to understand is how the work of testing can be performed in a complex and volatile situation, and how the testing can be improved to give the results of testing a stronger impact in the design and development processes, and thereby improve the quality of the product. On the basis of these preliminary questions, we have performed a number of limited case studies.

Although the strategy that has been used is that of case study, it has been supplemented by strategies and techniques taken from other areas. For example, the data has been analysed in a fashion that is based on a grounded theory approach. The data have, as already mentioned, been collected using many different techniques from many different sources. The techniques have included observation, interviews, and studies of documentation and project models.

Since we work in a software engineering context, the method of presenting the results can differ from a traditional account of a case study, or ethnography, where results are often presented as rich descriptions. Here, the object of the report is not to allow the reader to experience being there, but is to make the results accessible to readers within the field of software engineering.

Yin states that research design is the logic connecting the data to be collected to the research questions, and the design should indicate what data are to be collected, and how the data are to be analysed. Five components are particularly important, but the two final components are those that are least well developed in case studies ([61], p. 21):

- The study questions, often in terms of "who", "what", "where", "how" and "why", affect the design that is appropriate to the study, and a case study approach is most suitable for "how" and "why" questions. These questions can also be answered by experiments and histories. "What", "who" and "where" questions are more likely to favour surveys or an analysis if archival records ([61], p. 6).
- The study propositions direct the attention to what is within the scope of the study: the "how" and "why" questions that point towards the object of the study. There may be reasons that the study will not have any propositions, such as in a situation where the object of the study is exploratory, although even such studies must have a purpose, and criteria by which the success of the study will be judged.

- The unit of analysis is related to the problem of defining what the "case" is. In a classic case study, this can be an individual. It can also be an event or entity that is less well defined than an individual. The definition of the unit of analysis is related to how the research question is defined.
- The data should be logically linked to the propositions of the study
- Criteria must be set up for interpreting a study's findings.

As already mentioned, we have studied how the testing is performed, and then studied why it is performed in this way, in order to both understand, and find ways of improving it. The fact that our study has been based on "how" and "why" questions, combined with the fact that we have worked with action research, has meant that we have worked with a case study approach. The unit of analysis here has varied according to the focus of the different case studies that have been part of the larger research project, but in general they have been on the level of a small group, the individuals who have been involved in designing and performing the usability testing, whilst the remainder of the organisation has been part of the context where the study has taken place.

In the research in question, the final two activities were performed by using a grounded theory approach. For more information on how this was done, see Section 2.5.

Yin presents a number of criteria that have commonly been used to establish the quality of empirical social research, which are all relevant to case studies. These criteria are not only important during the design phase of the study, but should also be applied during the conduct of the case study. They deal with construct validity, internal validity, external validity and reliability ([61], pp. 35-39).

Construct validity deals with establishing correct measures for the concepts being studied, and is especially problematic in case study research. Three tactics are available to increase construct validity. 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, as already mentioned used many different sources of information. Ensuring a chain of evidence has been part of the data collection and analysis process, where a "study database" has been maintained. This is mainly in the form of a computer based document, an account of the study, containing such things as records of meetings and other activities performed in the study, transcriptions of interviews and observation notes, and records of documents and articles that are relevant to the study. The audio recordings are also stored digitally in mp3 format. The written document contains marginal notations of thoughts and ideas that arise 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, where we present the results in a fashion that allows the reader to follow the data collection and analysis process.

The most important research collaborators within the industrial organisation, those who are directly involved in the testing efforts, have been an integral part

of the study, and have therefore been closely involved in many stages of the work. They have therefore been available for testing thoughts and hypotheses during the study, allowing frequent opportunities for member checking. They have also been involved as co-authors in the process of writing articles for publication, which has also meant that member checking has been an integral part of the research process. One further method for ensuring validity has been through discussions together with the other members of the research group, where material has been presented, giving colleagues the chance to react to the analysis and even suggest and discuss alternative explanations or approaches.

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. Specific analytic tactics to ensure internal validity are difficult to identify, but 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 process of analysing data according to some of the principles of grounded theory is one method that we have used in an attempt to ensure that issues of internal validity are dealt with.

External validity concerns knowing whether the results of a case study are generalisable outside the immediate case study, and this 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, however, the investigator tries to generalise a set of results to a wider theory. However, generalisation is not automatic, and a theory must be tested by replicating the findings, in much the same way as experiments are replicated. Since the research process as presented here is based on a study of one company in a limited context, it is not yet possible to make claims about the external validity of the study. What we can say is 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. What remains to be done is further work to see how applicable the theory is for other organisations in other or wider contexts.

Reliability is a familiar concept, and 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 ([61], 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 ([61], p. 69). As mentioned previously, when discussing construct validity, a case study database has been maintained, containing the most important details of the data collection and analysis process. This should ensure that the study is theoretically replicable. One problem regarding the replicability of a study such as this, 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, which is an important factor to take into account.

For more details of how the study at hand relates to the principles and methods presented above, see Section 2.6. Before we look at this we take a brief look at the theories and principles of grounded theory.

2.5 Grounded theory

Grounded theory (GT) is both a strategy for performing research and a style of analysing the data that arises from the research ([59], p. 191). There are some typical features of a grounded theory study. It is a systematic but flexible research style that gives detailed descriptions for data analysis and the generation of theory. It is often interview-based and is applicable to a large variety of phenomena ([59], p. 90). However, even though interviews are an important source of information, other methods such as observation and document analysis can also be used ([59], p. 191). In our study we have not attempted to work according to pure GT practice, and have instead applied a case study perspective, using ethnography and participatory design. However, our study has relied to a large degree on the same basic methods of collecting data, although observation has perhaps had a more central role than interviews.

The value of the methodology is that it not only generates theory, but also grounds the theory in the data, where the interpretation is based on systematic enquiry ([63], p. 8). In grounded theory, data collection, analysis, and theory development and testing are carried out concurrently, and it is particularly useful in areas where theory and concepts that can explain what is going on are lacking ([59], p. 90). Strauss and Corbin use the term to mean that theory is derived from data that is systematically collected and analysed through a research process where the creativity of the researcher is an essential ingredient, and that data collection, analysis, and the theory formed are closely related to one another; a researcher starts with an area of interest and the theory arises from the data. Since grounded theories are based in the data, they are likely to offer insight and enhance understanding, and be a meaningful guide to action ([63], p. 12). The rapid pace of change in the industrial context where we are working means that theory and concepts concerning the field are in a state of change, and GT is therefore a suitable methodology for us to use. The fact that we have had easy access to empirical data has given us access to the data necessary to perform this type of analysis in an iterative research process.

The development of theory, which Strauss and Corbin call theorising, is often a long and complex activity. It involves not only conceiving ideas (concepts) but also formulating them into a systematic explanatory scheme ([63], p. 21). Here, theory is: "… a set of well developed categories (e.g. themes, concepts) that are systematically interrelated through statements of relationship to form a theoretical framework that explains some relevant social, psychological, educational, nursing or other phenomenon" ([63], p. 22). The foundation of the approach to theory is that it should be emergent, and a researcher must enter an area without preconceived concepts, an existing theoretical framework, or a well thought out design. These are instead supposed to emerge from the data. ([63], p. 33).

Robson lists a number of attractive features and problems with grounded theory ([59], p. 192). The attractive features are that: it provides explicit procedures for generating theory; it is a research strategy that is systematic and coordinated whilst still remaining flexible, and it provides explicit procedures for data analysis; it is useful in research areas where the theoretical approach is not clear or does not exist. Some of the problems are: it is difficult to start a study without pre-existing assumptions or theoretical ideas, which is what is called for in the original formulation of grounded theory, according to the principle of emergence; there may also be tension between the evolutionary style of a flexible study and the systematic approach stipulated in grounded theory.

In accordance with what is written above, a conflict with one of the basic principles of GT arises in our case. Since we work according to an action research agenda, where our express intention is to make improvements in the testing procedures within the company, we are not entering the field without any pre-existing assumptions. We have also found that the flexibility of the study has led to changes in focus, as facts and factors emerge that we find to be relevant and interesting for our study. However, we have seen the advantages of using the systematic procedures associated with GT for generating theory as being an asset in our study.

One problematic area in this type of study concerns the goal of always attempting to work on the basis of the members' point of view. The problem is that the members are not always aware of what is actually going on in the workplace (see Section 2.7 for more information regarding this, and how ethnography reconciles this problem). Relying completely on the members' point of view can give a biased view of the situation as it is in reality. The problem is to capture the members' point of view, whilst at the same time analysing the context to verify the accuracy of that point of view and compare it with our picture of the same context. By combining the use of techniques taken from the field of ethnography with an iterative grounded theory approach to the analysis, refining the results of the analysis, and then checking the results of the analysis with the participants in the study is one way that we have seen of combining the two points of view; those of the researchers and those of the practitioners. In this way we reach a balanced view of the situation and context.

Grounded theory, although it is often characterised as a qualitative research methodology, can also include the use of quantitative data ([59], p. 192). Strauss and Corbin state that the aim of theorising is to generate theory, and that both qualitative and quantitative technology are a means to do that ([63], p. 27). Combining methods is not specifically about triangulation, but is more concerned with using all methods at the disposal of the researcher to develop a well formed and comprehensive theory ([63], p. 33). As concepts and relationships emerge from the data, the researcher uses these to decide how to go about gathering the data that furthers the development of the theory. To do this, it may be necessary to make use of quantitative or qualitative methods ([63], p. 33). The qualitative and the quantitative should direct one another in a circular and evolving feedback process, where each contributes to the theory in its own particular way ([63], p. 34). The main data in our study has been qualitative, and they have been analysed in a qualitative fashion, although in one of the case studies (see Chapter 6) we have also collected data that have been analysed in a more quantitative manner.

2.6 Collection and analysis of the data

Here, we examine first how data is collected, in both case studies and grounded theory, and how this compares to the approach taken in our study. Following this we examine how the data is analysed, again from a case study and GT perspective, and how our study compares to these theories and practices.

Case study: In a case study, there are many sources of evidence; the most commonly used ones are documentation, archival records, interviews, direct observations, participant-observation, and physical artefacts. A good case study will use as many of these sources as possible, since all sources have their inherent strengths and weaknesses, and all sources are complementary ([61], p. 85-97).

Documentary evidence can take many forms, and is likely to be relevant to every case study topic. Archival records often take the form of computer files and records. The advantages of the above are that they can be viewed repeatedly, are exact, are not created as a result of the case study, and have a broad coverage of time, events and settings. Disadvantages are that they may be difficult to retrieve, and can even be deliberately blocked, can exhibit bias if collection is incomplete, and can reflect the bias of the author. Archival records in particular can be precise and quantitative, but they can also be difficult to access for reasons of privacy. In the study, we used a number of sources of documentary evidence, all of which were freely available to us. These included general documents concerning the project models in use at the company, and the organisational structure. Documents that were more specifically connected with the testing process were, for example, the forms and research protocols used in the testing procedure (see Appendix A for examples), presentations used when presenting and discussing test results with management and customers, and spreadsheets where the data from the testing was stored. There was an open climate that allowed us easy access to the documents that we needed.

Interviews are one of the most important sources of case study information, and are likely to take the form of guided conversations rather than structured queries. The advantages of interviews are that they are targeted, focusing directly on the case study topic, and provide insights. Case study interviews are commonly open-ended, but can also be focused, or structured along the lines of a survey. Some of the disadvantages of interviews are that they can lead to bias if the questions are improperly constructed, that responses can be biased, that things can be inaccurately recalled, and that the interviewee says what she or he thinks the interviewer wants to hear. Interviews should be considered to be verbal reports only, and the information should be corroborated with information from other sources in a process of triangulation. Interviews are a large source of the data in this study, and have mainly been in the form of open ended interviews, but also as more structured interviews. Section 2.9, concerning the case studies that are part of the thesis contains more details on this. The large number of occasions that interviews have been performed, and the process of refining the interview questions according to the theory that has emerged in the analysis process has meant that the information gathered from interviews has been checked and triangulated. The interview material has been triangulated by observations, and document studies, leading to confirmation of findings, or further avenues for exploration.

Direct observation and participant observation have the advantage of covering real events in real time, and give access to context. They can however be time-consuming, selective, reflexive (events may proceed differently because they are being observed) and costly. Participant observation gives insight into behaviour and motives, but can be biased due to the investigator manipulating events. Observations can range from formal to casual data collection activities. In participant observation, the investigator is not merely a passive observer, but may actually participate in the events being studied. Observations by the researcher have been a central part of this study, ranging from observations of the testing procedure together with a test leader and a tester, participating as one of the testers in one of the test cycles, to participating in discussions, seminars, presentations, and project meetings. More information can be found in the Chapters 5 and 6, detailing the specific case studies, and Section 2.9. The observation process has been as unobtrusive as possible, and has been so well established, at least within UIQ, that is has been an accepted part of the everyday cooperation between the university and the company. This means that there is a diminished risk of the observations and participation having negative effects.

Physical artefacts give insight into cultural features and technical operations, but there can be problems of selectivity and availability ([61], pp. 85-97). The physical objects that have been available and studied have largely coincided with the forms and research protocols used in the testing procedure, as mentioned earlier. The testing itself has not relied on physical artefacts apart from the devices to be tested.

Grounded theory: In grounded theory, it is particularly difficult to separate the process of collecting data from the process of analysing data, since these two

activities are concurrent and dependent on one another. This differs from the traditional linear model of research, where all data is collected, and then analysed. In a grounded theory study, the researcher should make several visits to the study to collect data, and this data should be analysed between the visits. Data is gathered until the categories found are saturated ([59] p. 192). Theoretical saturation is the point in category development where no new properties, dimensions or relationships emerge from the analysis ([63], p. 143).

In grounded theory, there is no element of gaining a representative sample, and no notion of random sampling from a given population. Sampling takes place on the basis of gaining information about the emerging conceptual categories ([59], p. 193). Data is sampled in a purposive manner. This means that the people who are interviewed or observed are chosen, and a selection of data is built up, to satisfy the researcher's needs, according to the researcher's judgement as to relevance. In grounded theory this is referred to as theoretical sampling ([59], p. 265). This type of theoretical sampling has been used in this study, choosing the appropriate participants in the study, according to the purpose of the particular phase of the study. Some participants, such as the person who is mainly responsible for the performing the testing, and the head of the interaction design department, have been central figures, involved in almost all stages of the study, whilst others have been less frequent participants, and have been called upon as the need arose, to gather new information or explore associated factors.

We now look more closely at analysis as it is performed in case studies and grounded theory.

In studies where qualitative data is involved, there are no conventions for analysing the data corresponding to the methods used when analysing quantitative data. There are, however, ways of dealing systematically with qualitative data, and in a study of this kind, where the methods that generate qualitative data are a substantial part of the study, serious attention must be given to how they are analysed ([59], p. 456).

One analysis typology that is closely linked to the method of data analysis used, and where there is a progression from more to less structured ([59], p. 457), is:

- Quasi-statistical methods, relying on the conversion of qualitative data into a quantitative format. These use word frequencies and correlations to determine the importance of terms and concepts.
- Template approaches, where key codes are derived from theory or research questions or from initial readings of the data. These codes become templates for data analysis. Text segments that are empirical evidence for categories are then identified.
- Editing approaches are more flexible than the above, and there are few predetermined codes. Instead, the codes are based on the researcher's interpretation of the patterns or meanings in the text. This is typical of grounded theory approaches.

- Immersion approaches use methods that are fluid and not systematised, and are the most interpretive and least structured. They are seen as difficult to reconcile with a scientific approach, and are close to literary or artistic interpretation.

In our study, the approach taken is closest to that defined above as an editing approach. No codes were specified at the beginning of the study, and the codes that arose appeared as a result of the iterative research process, and the annotation process. Analysis has been done by annotating the case study database, where the most important element is the research logbook (see Section 2.4 for further details regarding the contents of the database) and thereby finding emerging themes, codes and categories. In one of the case studies (see Chapter 6) a trial version of NVivo 8 [64] was used as a tool for storing, analysing and categorising data. The results of this process of analysis and reflection have also been written into the research log book as a way of summarising and clarifying the findings so far.

The codes and categories that have emerged from the data have been used to refine the research strategy and questions, and point out new areas of interest in the study, in an iterative process. The material in the case study database has also been available for repeated readings for re-evaluation, reanalysis and reinterpretation as new ideas have emerged.

Strauss and Corbin write that qualitative analysis is not a case of quantifying qualitative data, but is a non-mathematical interpretation whose purpose is to discover the concepts and relationships that exist in the raw data, and then organise the data into a theoretical explanation ([63], p. 11). The work of analysis can be described as a typical series of steps. The first of these is giving codes to the initial set of materials, obtained from interviews, observation etc. The next step involves adding comments and reflections (commonly known as memos). The material is then studied, to discover similar patterns, phrases, themes, and relationships. The focus of the next phase of data collection is then influenced by the patterns and themes found above. Gradually a set of generalisations is elaborated that can explain the consistencies found in the data, and these generalisation are linked to a body of knowledge in the form of theories ([59], p. 459). This general process described here has characterised our study. As previously detailed, the research material has been collected in the case study database, and the annotation process has consisted of marginal annotations. The thoughts and themes that have arisen in this process have been the basis of and have affected the continued research. After each step in the analysis process, the earlier material has been read to see if support can be found for the new ideas and theories, and at the end of the process, a theory has been formulated that is grounded in the material.

The aim of analysis in grounded theory is to generate a theory that explains the data. It aims at finding a core category that is at a high level of abstraction that is still derived from, or grounded in, the data. This is done by finding categories in the data, finding relationships between the categories, and conceptualising and accounting for the relationships by finding core categories ([59], p. 493).

The repeated comparison of collected data and emerging theory is called the constant comparison method of data analysis, and the process of data analysis is systematic ([59], p. 193). In GT, the analysis is done by carrying out three kinds of coding. These are open coding, axial coding, and selective coding. These coding processes need not be sequential, and are likely to overlap ([59], p. 493).

Many analyses of flexible design studies have been influenced by grounded theory, but whilst some of them follow the detailed descriptions laid down in the grounded theory methodology, many others are more "in the style" of grounded theory ([59], p. 492). The iterative approach found in grounded theory where data analysis fuels data collection, which then leads to further analysis, is a feature of the analysis of most qualitative studies ([59], p. 487). One viable approach in case studies or ethnographic studies, is using the techniques from grounded theory in a relatively relaxed manner, not necessarily using the terminology of open coding and axial coding ([59], p. 487). In our study we have used the techniques from GT in this type of relaxed manner. We have not explicitly followed the techniques called for according to the GT tradition, but in our manner of working, we have in principle followed the mechanisms of open coding, axial coding and selective coding.

Open coding is defined by Strauss and Corbin as the analytic process for identifying concepts, and their properties and dimensions, from the data. This process leads to a structure for building theory ([63], p. 101). Open coding is about interpretation, and finding the theoretical meaning in the data. It entails splitting data into discrete parts that seem to be units of data (these may be sentences of paragraphs), and asking what these pieces of data may be examples of. In our study, this has been performed directly in the logbook, as previously detailed in this chapter. A code, or label, is applied to the unit. These labels are at first provisional and can be changed, and one piece of data may have several codes if it appears to fall within several conceptual categories. The notation that has been recorded in the margins of the research logbook is our equivalent of these codes and labels. Analysis begins before data collection is complete, and this has also been true in our case, where the research cooperation has been so iterative. In open coding, it is important to bear in mind ideas that arise about relations between categories, and thoughts about a core category. In our case, these types of reflection have been recorded in the logbook in the passages where the results of analysis and reflection are written as a summary and clarification of the findings so far. The goal is to reach a stage where the categories are saturated, where continued analysis no longer adds to the concepts, categories and insights that you have already reached ([59], p. 494).

In axial coding the categories are related to their subcategories. Here, categories are related to their subcategories to form explanations about phenomena ([63], p. 124). Axial coding, also called theoretical coding, involves linking the categories that have arisen during open coding. The basis of axial coding is recombining the data that has been split apart into categories by open coding. It involves building a model of the phenomena. It is still asking questions of the data, but the questions are focused on the relationship between the categories ([59], p. 494).

This type of analysis is also related to the reflections and results of analysis as they are recorded in the logbook. It is also connected to the re-readings and continual analysis of the previous research data, where codes and categories can be reanalysed and reformulated in accordance with the new information and hypotheses. In this process, we are still working within the principles of ethnography, and are attempting to understand the situation from the members' point of view. However, in combining different elements to a whole, we can reach insights that may be beyond the members' point of view, since we see things from angles that may not be accessible to the members, and which they may only see fragments of. At the same time, we are still attempting to understand the fragments from the members' point of view. This demands the skills of the ethnographer, where the researcher bases his or her knowledge on personal experience, and the ability to take a dispassionate and holistic view of the studied context.

Selective coding is the process of integrating and refining the categories, as it is not until the categories are integrated to form a larger theoretical scheme that the findings can be called a theory ([63], p. 143). In selective coding, one aspect is selected and focused upon as the core category. This category arises from the axial coding, that leaves you with a picture of relationships between different categories, giving a feeling for what the study is about, allowing you to understand the overall picture. In grounded theory there must be one central focus to the elements that remain in the study. If there is more than one, they should probably be integrated into a single concept at a higher degree of abstraction that still remains grounded in the data. The core category is the centrepiece of the study, and is the theory that allows you to understand the study as a whole. In our case, this final step has not been a process of recoding, but has rather been a process of summarising the results of the study, and checking the theory together with colleagues and the practitioners involved in the study. Another part of this process has been the cooperative writing process, where the theories that the researcher, together with the practitioners, has arrived at are once again formulated and refined to reflect the situation that we have studied.

To summarise the above very briefly, we have worked in an action research tradition within the framework of Cooperative Method Development, which is an approach to empirical research consisting of three phases; understanding practice, deliberating improvements, and implementing and observing improvements. In our application of CMD, we have used a case study approach and analysed the data according to the principles of grounded theory.

In our series of limited case studies, we have examined the theory and practice of usability testing in the industrial setting, in a study that has been a combination of exploratory and explanatory. The data collected in the study have been both qualitative and quantitative, with many different sources of data being used. The industrial participants in the study have been closely involved, partly in the process of deciding what should be studied, and partly in the data analysis process, giving their views of what is actually happening within the context where they are working. The data analysis has been performed

according to the principles of grounded theory, allowing the development of theory grounded in the data collected in the study, although the coding activities stipulated by GT have been only loosely adhered to.

Now, having looked more closely at CMD, the overall research principle that the study is grounded on, at the case study approach that has been used throughout the research cooperation, and at grounded theory, which has influenced the data analysis procedures, we now proceed to look more closely at ethnography and participatory design, which also have been central themes and tools in the research process.

2.7 Ethnography

The studies that have led to this thesis have been inspired by ethnography, which is a research strategy taken from sociology, and that has its foundations in anthropology [65]. 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 [65]. Ethnography has often been closely connected with the field of Computer Supported Co-operative Work (CSCW) where it has become the dominant research methodology used to study workplaces [65], but has also become one of the tools and techniques of participatory design (PD), where the connection to ethnography is the commitment to involve participants in the study and where the participants' knowledge of their own practices is valued [65]. The focus of ethnography has moved from investigations of small scale societies to the study of specific settings in industrialized societies, and it is an approach that is found in most of the social sciences, and in interdisciplinary fields such as HCI and Human Factors Engineering [65].

Rönkkö characterises ethnography as the study of work as it occurs, without the imposition of specific research questions on the participants [65]. It produces a rich and concrete portrayal of the situation that is studied and describes a setting as it is perceived by those who are involved in the setting [65] – the point of ethnography is that is captures the members' point of view. An explanation of "the members' point of view" is provided later in this section.

According to Rönkkö, the motivation for creating an ethnography is that "things are not what they seem" and that appearance do not tell the whole story. Thus it is necessary to look behind appearances in a detailed and systematic way – which should be the inspiration of all scientific work – and it is the job of the ethnographer to collect the evidence and set it in a framework. The job of the ethnographer is not only to write up what has been found in the field, it is also to interpret and analyse this material [65].

Ethnography, in much the same way as participatory design, has not concentrated on creating a single method, supported by tools and techniques. There are however a number of methods that are in use within the field, that have been developed to enable the achievement of a "descriptive and holistic view of activities as they occur in their everyday setting from the point of view of study participants" ([66] p. 967). The methods used include observation, interviewing, self-reporting techniques, remote data collection, artefact analysis

and record keeping [66]. In our studies, the main methods have been observations and interviews, combined with literature studies of the different areas of interest.

Ethnographic practice is based on several basic principles [66]:

- Natural settings
- Holistic
- Descriptive
- Members' point of view

Natural settings. Even though studies can take place in situations that remove people from everyday settings, or change those settings, the point of studying activities in their natural settings is that people have a limited ability to describe what they do or how they do it without having access to the everyday aspects of their lives, and that some aspects of peoples' experience can only be understood through observation of activities as they occur [66].

Holistic. Activities must also be understood in the larger context in which they occur (this is also related to the need for natural settings) and studying an activity in isolation, without reference to other activities that it is connected with can give an incomplete and misleading understanding of the activity [66].

Descriptive. Ethnographic accounts have always been descriptive rather than prescriptive, and ethnography is primarily concerned with understanding events as they occur, rather than evaluating everyday practices – even though ethnographic accounts can be used to point out changes or problems, based on an understanding of the situation as it is [66].

Members' point of view. Ethnographers strive to see the world from the point of view of the people who are studied, and try to describe the world in terms that are relevant and understandable for the study participants [66]. An ethnography should "present a portrait of life as seen and understood by those who live and work within the domain concerned" ([67] p. 430). This is what is known as the members' point of view.

However, ethnography emphasizes a detailed understanding of a culture, originating from intense long-term involvement where the ethnographer immerses him/her self in the culture in question. The aim of the ethnographer is to describe the studied situation without erasing the complexity and the studied people's point of view. A result of this intimacy is the subjective understanding necessary to overview, correlate and integrate the various ways in which things appear in the field. It provides the means to discover and represent "a way of life". An ethnographer knows things in ways others do not and cannot know, because they partly base their knowledge on personal experience. Although it is important to work on the basis of representing the members' point of view, it is also important to realize that the members of the culture that is studied themselves lack the entire picture of correlations and evidences that the ethnographer produces, and also the professional skill to distance themselves from themselves [65].

Looking at the above in the context of our study, we find that by having access to the organisation and being accepted as a natural part of the cooperation between the university and the company, it has been possible to take a holistic view and study the context in which the work is performed. The climate of cooperation has been so open that it has even been possible to participate in the cooperative work and discussions that took place together with an industrial partner in England. This cooperation led to further developments in the test methodology, and it was possible to observe some of the processes that led to these developments, and draw conclusions regarding the testing and analysis process.

A focus on working in and studying the natural setting has been a central part in the research cooperation in this study. The study has consisted of periods of being active within the everyday work of the company, observing testing activities, participating in meetings, and observing cooperation with colleagues, and even cooperation with external/partner organisations. Further periods have been spent analysing the research material and using the results of the analysis to return to the industrial environment with new ideas and knowledge, which have been used to further the cooperation and the understanding of what is happening in the setting. This mix of models has given depth to the studies and has led to developments in the testing methods in use.

We have worked with ethnographic methods, and have been concerned with understanding the events in the everyday practice of the practitioners in the study. However, we have also, in accordance with the action research agenda, gone further than the descriptive side that is characteristic of ethnography. The ambition was thus not only to observe and understand practice, but was also to make improvements in the way of measuring and reporting usability, and using the results in the development processes. We have not only attempted to see the situation as it is, we have also been concerned with the situation as it could be in the future, and this has affected both our working methods and style of reporting the results. We have for example led meetings and seminars where the intention was to introduce new ideas from the academic environment and other industrial contexts. This was to stimulate discussions of how the evaluation framework could be improved, and even lead to concrete changes. This is also reflected in the contents of this thesis, which are not presented in the form of an ethnography.

The fact that the work we have performed is aimed at the software engineering community is another reason why we have chosen not to present the work as a traditional ethnography. Instead we present our work in a form that, although it is based on a certain mindset and way of working, is adapted to the software engineering discourse, in order that our work can find acceptance in that forum, and contribute new knowledge that can be of use to the researchers and practitioners in that field.

Another way of ensuring an emphasis on the “members’ point of view” is by conforming to the principles of PD. In the next section we take a closer look at PD, how it has influenced us and how we have used it in our study.

2.8 Participatory design

In the cooperative development of the usability test method, we have worked with a PD perspective. PD is an approach towards computer 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 [68]. Due to the diversity of perspectives and background of those who are practitioners, there is no one single understanding of what PD is. However, there is a shared view regarding the core principles of PD [69] :

- Every participant is an expert at what they do
- The voices of all participants must be heard
- Good design ideas arise when participants with different backgrounds collaborate
- It is better to spend time in the user's environment rather than performing tests in laboratories
- Group participation in decision making
- Individual and group empowerment
- The purpose of participation is not only for reaching agreement but also to engage participants in adaptation and change of their environment.

In relation to the above principles, the role of the practitioners and the depth of their experience have had a central position in our study. Based on their experience in the field, and knowledge of the industrial environment in which they operate, the practitioners have strongly influenced the direction the research has taken. The mix of participants, from interaction designers and architects, test leaders, technical writers, to persons in management positions, has given us access to a broad base of knowledge, of both the company itself, and of the area in which the company is active. Many of the people who have been central in the study also come from a background where they have developed an interest in and knowledge of qualitative methodology and ethnographic methods, and the principles of participatory design.

The research methodology in use has given participants the chance to make their voices heard. This has been made possible by the mix of methods that has been in use in the research, from interviews and observation to participation in project and management meetings and presentations and discussions of test results with different stakeholders. This has allowed staff at many levels within the company to make their voices and opinions heard in the research.

This mix of people with different backgrounds and roles has also fertilised the research effort, by giving us access to many different points of view and fields of experience. For example, in many of the discussions, the group has consisted of researchers, the head of the interaction design department, the test leader, and a technical writer who also had a background as mathematician and engineer. Most of these people had been working together for a long time. This mix of

people and the familiarity they had with one another ensured an open climate where a multitude of ideas were discussed.

The main part of the research has taken place in the user's environment. This was partly due to the fact that the researcher had access to his own workplace at the company, but also because he has been able to follow at close quarters the process by which the testing methodology has been developed, the actual testing that took place, and the cooperation between different companies involved in the development process.

The group of people that has mainly been responsible for developing the evaluation framework and disseminating the results throughout the organisation has a long history of working together and of utilising one another's skills and competencies. This, taken together with their backgrounds, has meant that decisions have been reached in an open climate and in a democratic fashion, based on the knowledge and experience of the participants. The company itself, which has grown from a small company where many of the people who were originally employed had a background in qualitative methods, and who have been influenced by principles of PD, has also recognised the importance of the work done by the team, and has given the team freedom to develop the test as they have seen best, providing that the results have been sufficiently convincing to the stakeholders who have had an interest in the results. This situation also points to the fact that the group has been empowered, as individuals and as a group, to make decisions about the best way to design the test.

All of the above, taken as a whole, show that the people who have been involved most closely in this study have, as individuals and as a group, been engaged in the decision making process, but have also been empowered to adapt and change their working environment, meaning that they have had the possibility to develop and perform usability testing in a way that is in accordance with their beliefs, and in accordance with the principles of the importance of qualitative methods, ethnographic methods, and participation.

Researchers in the field of PD are concerned with the conditions for user participation in the design and introduction of computer based systems in the workplace [70]. Three main issues that have dominated the PD research within PD are, according to Kensing and Blomberg [70]:

- The politics of design
- The nature of participation
- Methods, tools and techniques for use in design projects.

The politics of design. PD researchers have always been explicit about the political aspects of the introduction of computer-based systems. PD began in the 1970's as a reaction to the ways that computers were introduced in workplaces and the negative effects that this had on workers. The neglect of workers' interests was at the centre of critique and PD researchers argued that computers were becoming a management tool to exercise control over workers. Researchers attempted to build proficiency amongst workers to strengthen their bargaining power. Later changes in workplace politics forced a change in some

of the assumptions that characterised the work of PD researchers, and focus moved to the rationales for participation and the way different actors in an organisation could influence the design and implementation of technology [70].

Participation. Three basic requirements for participation are: access to relevant information; the possibility to take a position on problems, and; participation in decision making. The participation of the users of technology is seen as a precondition for good design. The likelihood of systems being useful and well integrated into organisational work practice is thought to be increased by taking into account the skills and experience of workers. PD researchers hold that designers need knowledge of actual work practice and workers need knowledge of technological options, and these types of knowledge are developed best if workers (and other members of the organisation) and designers cooperate in design projects. A shift is taking place in who takes place in PD activities, and various people throughout the organisation, including management representatives, are nowadays included in PD activities. In many cases it is not possible for all those who are affected by the design effort to participate, and in this case the choice of representatives and the form of their participation must be negotiated together with management and the workers themselves [70].

Methods, tools and techniques. Developing one single PD method has never been the aim of PD researchers, although some groups have systematically organised their design practices into collections of tools and techniques. Developing tools and techniques is important in PD projects, and they have become a part of PD researchers' repertoire for action. The tools and techniques are designed to encourage a practice where researchers and designers are able to learn about users' work, that allows users to take part in technology design, and where both technology and work organisation are in focus. Tools and techniques for work analysis, such as workshops, workplace visits, and, increasingly, ethnographically-inspired fieldwork techniques are complemented by others that focus on system design, including scenarios, mock-ups, future workshops and cooperative prototyping. All PD tools and techniques have in common the aim of allowing users and designers to connect work practices with new technologies [70].

When we place the above in relation to our study, we find that the work done here reflects the movement away from the political. The focus of the work performed here has not specifically been aimed at reducing negative effects of computerisation, or improving the control that the workers have over their working conditions. The people involved in the project already have a good deal of autonomy and work within an organisation where participation is encouraged and appreciated. There has thus been no specific political reason for encouraging worker participation, apart from a general principle that those who are involved in the everyday running of an operation should have the possibility to exercise some control over their everyday activities. We have seen the PD process and the process of cooperation as it has taken place in the study as contributing to building competence within the organisation but also see this as being mirrored by the increase in the knowledge that can be brought to the academic area.

The main reasons for encouraging participation have therefore been to utilise the knowledge of the participants, to improve the quality of the work that they can perform, to increase the value of that work for the rest of the company, and to learn from the everyday work processes that are taking place in this new and rapidly changing industry. The principle behind this study has thus been to use the skills and knowledge of those who are most closely involved in the work, in order to design and implement a solution that is well designed, suitable for its purpose, and suitable in the organisation where it is to be used. These skills and knowledge have been both in the operational area, and in the area of organisational knowledge. This has also led to a situation where the range of roles that have been included in the study has been extensive. Although the main part of the study has been performed with the people who are responsible for the everyday running of the test program, and analysis of the results, participants have also included management representatives, who have participated as interviewees and as participants in meetings and discussions that have taken place during the design and use of the evaluation framework.

As presented in this chapter and in the rest of the thesis, the range of tools and techniques used in this study has been extensive. The main intention of our efforts has been to capture and include the "members' point of view" in the study. The main focus has been on those who work "on the shop floor", although is has been seen important not to exclude participants from management levels. All methods that have been seen as suitable for capturing and including this members' point of view have therefore been seen as being useful, and as candidates for use in the study.

The combination of methods that have been in use in the research, and its combination of ethnography, and PD, is also reflected in the evaluation framework itself. We have not specifically taken this up in this chapter, but we return to it in Chapter 4, dealing with the design of and philosophy behind the evaluation framework, regarding the way in which mobile phone users are involved in the testing procedure.

In the remainder of this chapter, we present in brief the methodology as it has been used in the different projects that have been performed as part of this study.

2.9 Empirical methodology in Chapters 5 and 6

This work presented in this thesis is all part of the long-term cooperation between BTH/U-ODD and UIQ Technology, mainly together with members of the interaction design team, which has centred on the development and evaluation of the test package. The prime area of interest has been on creating a method for quality assurance, on developing key performance metrics to measure user experience and on the combination of qualitative and quantitative results.

The overall research question for the study has been how to develop and evaluate an evaluation framework for mass market mobile devices, measuring usability and quality in use, allowing measurement, comparison and

presentation of the usability of mobile devices. This involves solving the problem of how to capture adequate real world usage needs in a continuously changing society, and how to support different information needs in the cooperative process of building the software.

Chapters 5 and 6 present different aspects of this overall research question, dealing with: Chapter 5: Balancing demands for agile results and plan driven results; Chapter 6: Based on our theories regarding the balance between agile and formal results, we investigated how stakeholders would prefer to be given the results of usability testing.

Chapter 5: Agile vs. Plan Driven

This case study in this phase of the research cooperation is based mainly on tests performed in cooperation between UIQ in Ronneby and Sony Ericsson Mobile Communication in Manchester. Here, the study focus changed, and the material was reanalysed and reinterpreted, and complementary interviews were performed to study new aspects that emerged and were found to be of interest. The research question in this case study was: How can we balance demands for agile results with demands for formal results when performing usability testing for quality assurance?

The primary data for this particular phase of the case study have been obtained through: observations; through a series of unstructured and semi-structured interviews [59], both face-to-face and via telephone; through participation in meetings between different stakeholders in the process, and; from project documents and working material. This working material included documents 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 researcher visited Sony Ericsson Mobile Communication in Manchester together with the test leader from UIQ, when the test leaders met to discuss the results of the testing and how they should be presented. During this visit, the researcher had the opportunity to interview both of the test leaders, both separately and together, and to participate in a meeting where the test leaders discussed the findings from the testing and the best way to present these findings for stakeholders at senior management level within Sony Ericsson. The interviews have been audio recorded, and transcribed, and all material has been collected in the research diary.

The diary is also the case study database, (see Sections 2.4 and 2.5 for more details regarding this) which collects all of the information in the study, allowing for traceability and transparency of the material, and reliability [61]. The mix of data collection has given a triangulation of data that serves to validate the results that have been reached. The primary material collected in this phase of the study has of course even been supplemented by material that has been collected in all of the phases of the research cooperation.

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 [59] 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. As previously noted, this particular case study is an extension of another parallel study. A drift occurred in the area of interest, as often happens in case studies [61], and the new research question arose. The first focus of the study was the fact that testing was performed in an actual industrial context, that was also distributed, and we were interested in the effect this had on the testing and the analysis and use 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, and the different types of stakeholders involved in the process, together with their information requirements. This became the area of interest in this case study. Looking back, this question has always been implicit in our research, even though it has not previously been made explicit. The question and the problem to be solved became more clearly defined as a result of participating in the workshop on software quality and other conference session at ICSE 2007, where these questions were raised many times and solutions were asked for.

We have tried to counter a number of threats to validity and reliability in the study. For a discussion of methods to ensure validity and reliability, and measures taken, see Section 2.4. One particular threat to reliability in this case study is bias introduced by the researchers most closely involved in the study. This has been addressed by cross checking results with participants in the study, and by discussing the results of the case study with research colleagues, which is in line with the validity internal checks that are recommended in ethnography. Another threat is that most of the data in the case study comes from UIQ. Due to the close proximity to UIQ, the interaction with staff there has been frequent and informal, and everyday contacts and discussions on many topics have influenced the interviews and their analysis. The interaction with Sony Ericsson Mobile Communication has been limited to interviews and discussions. Data from Sony Ericsson Mobile Communication 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, and in that way present a valid picture of industrial reality. This is in line with the recommendations for checking internal validity in case study procedure.

Chapter 6: Information Prioritisation

This case study is a direct continuation of the work that was done in the case presented in Chapter 5. There we found that there appeared to be two disparate groups of stakeholders who had differing information needs and worked according to different timescales. In this study, in order to verify the conclusions made in the previous case, we have examined how different stakeholders in the design and development process, who have an interest in viewing the results of usability testing, would prefer to be given the results of

the testing. We examine which presentation method will best satisfy their information needs. The presentation methods range from verbal presentations to visual presentations, and can contain both qualitative and qualitative data.

The work in the study was performed to answer the following five 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 (Designers & Product Owners) 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 the usability evaluation framework?
- 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 participants in the study were asked to answer a questionnaire and prioritise different methods for presenting test results. The participants were chosen by taking a cross section of staff within UIQ and two other companies, both of which were associates and customers. The participants were chosen in collaboration with the test leader from UIQ, and were all seen as candidates who had some kind of interest in viewing and using the results of usability testing.

The method whereby the respondents were asked to prioritize the presentation methods was based on cumulative voting [71, 72], a well known voting system in the political and the corporate sphere ([73], [74]), also known as the $100 test or $100 method [75]. Cumulative voting is a method that has previously been used in the software engineering context, for e.g. software requirement prioritization [9] and the prioritization of process improvements [76], and in [77] where it is compared to and found to be superior to Analytical Hierarchy Process in several respects.

In this method, each participant is given a fictitious sum of $100 to “purchase” ideas, and is given a free hand to use her or his money however they see fit. It is possible to distribute the money in any way amongst the features (or presentation methods in this case).

There are questions as to how many requirements Cumulative Voting can handle [9], but it has been used successfully in a situation where there were more than 20 objects to prioritize (see e.g. [78]). Problems may be caused by the fact that stakeholders lose overview as the number of factors to be prioritized increases [77]. In the case at hand, we only have 10 presentation methods that are to be prioritized, so this problem does not arise here.

Some issues have been noted with Cumulative Voting. It can be sensitive to tactical voting, meaning that it can only be used once in the same project, since the results of one round of voting can bias the participant's input in the next round of voting [75]. In our case, the prioritisation was only done once, so this was not an issue that we had to contend with.

It has been also found, especially in cases where there are many requirements to prioritize, that respondents have miscalculated points, so that the sum of points does not add up to the correct total [79]. To alleviate this problem, the tool we have used includes a function in the spreadsheet where the prioritization takes place, which calculates and shows the sum of the points allocated, and points out if there are points left for allocation or if the sum has been exceeded.

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.

In the same way as in the previous case study, we have been concerned with threats to validity and reliability. For a discussion of methods to ensure validity and reliability, and measures taken, see Section 2.4. To ensure reliability, we have ensured that the research procedures and data have been well documented via the research diary, which has functioned as the case study database. The results have also been discussed with research colleagues, to check the interpretation of the results. The data comes mainly from UIQ, meaning that it may be difficult to generalise the results to other organisations. Although Yin advises performing multiple-case studies, since the chances of doing a good case study are better than using a single-case design ([61], p. 53), this study has been performed as a single-case study. The case here represents a unique case ([61], 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. This case study has been performed to generate theory. 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.

One threat in this study is the scale of the study itself, and the fact that only ten people have participated. This means that it may be difficult to draw any generalisable conclusions from the results. Also, since the company is now disbanded, it is now no longer possible to return to the field to perform cross checking with the participants in the study. This means that the analysis is 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 small number of 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 can not be expected to accurately reflect the views of Product Owners.

To verify these results, further studies are needed. Despite the small scale of this study, the results give a basis for performing a wider and deeper study, and

allow us to formulate a hypothesis for following up our results. In line with the characteristics of the rest of the work performed as part of this research, we feel that the continuation of 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.

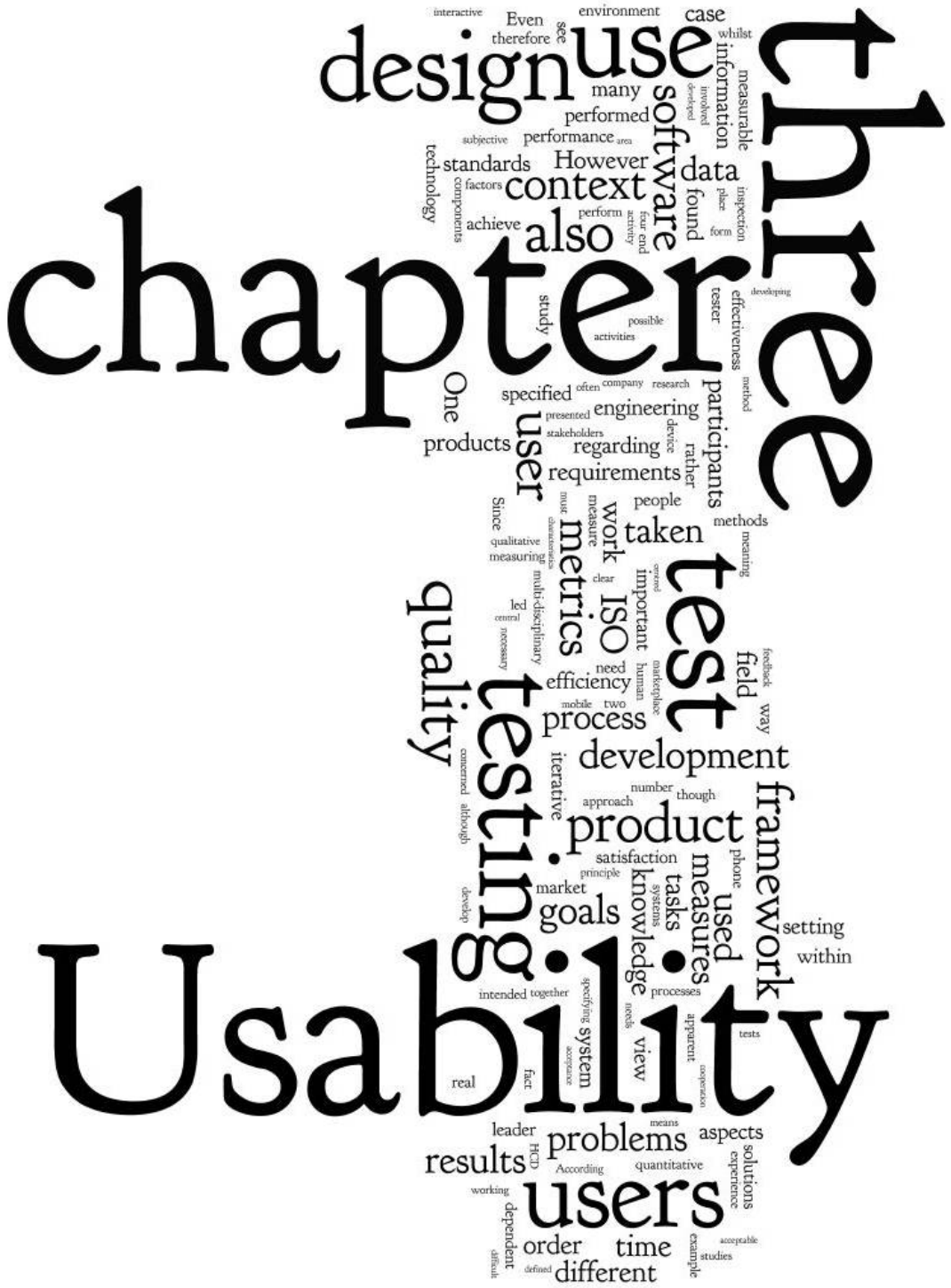
2.10 Summing up

To briefly summarise the above, the research performed in this study has been a closely woven cooperation between the academic and industrial partner. It has been based on an action research approach, following the processes of CMD, and has been based on a case study approach, where the data has been analysed using techniques taken from grounded theory. Ethnography and participatory design have been central to the research and cooperation approach

This particular mix of methods and techniques has been found to be a successful combination that has contributed to the development of a successful usability evaluation framework. The underlying principle in the research has been to capture and represent the "members' point of view", and this relates to both the participants within industry and, via the testing process, to the phone users who are engaged as participants in the testing procedure.

The research has been performed as an empirical study within industry. Such studies are relatively few in the field of software engineering, and it is seen as desirable for the area of software engineering to increase the number of such studies, and to carry out more action research and case studies, to improve the industrial relevance of software engineering studies [37].

In the following chapter we take a closer look at usability, the main area of concern in this thesis. We give an overview of how usability is defined, and how usability work is defined and discussed within different areas of research. We set this in relation to the work performed in this study.


chapter
three
Usability
design
use
test
testing
quality
metrics
ISO
framework
product
development
process
users
problems
results
goals
measures
knowledge
tasks
software
data
context
also
user
participants
requirements
regarding
products
engineering
specified
One
taken
work
people
methods
field
efficiency
need
important
multi-disciplinary
Since
qualitative
measuring
measure
iterative
approach
number
though
satisfaction
market
used
setting
within
intended
together
processes
system
view
real
fact
leader
aspects
solutions
quantitative
According
HCD
working
dependent
order
time
different
defined
example
studies
acceptable
standards
However
performed
many
information
measurable
Even
therefore
see
environment
case
whilst
interactive
subjective
performance
technology
factors
components
achieve
perform
found
form
inspection
place
four
end
activity
developed
involved
area
study
possible
activities
tester
effectiveness
developing
often
company
research
method
presented
stakeholders
device
rather
Since
must
characteristics
clear
current
led
central
necessary
human
marketplace
mobile
two
feedback
way
concerned
although
develop
principle
phone
systems
tasks
needs
specifying
acceptance
apparent
tests
cooperation
means
experience
difficult

Chapter Three

3. Usability – an overview

According to Dumas and Reddish, Usability is an attribute of a complete package, where changes in technology have blurred the borders between the different pieces of a product, which can include hardware, software, menus, manuals etc. [80]. Usability means that people who use a product can do so quickly and easily to accomplish their own tasks. It rests on four points, which are: focusing on the users; the fact that people use products to be productive; that users try to accomplish tasks, and; that users decide themselves if a product is easy to use. Usability must be built in to a product from the beginning, as it is affected by all design and development decisions [80].

Nielsen and Mack [81] state that usability is a broad concept referring to how easy it is to for users to learn a system, how efficiently they can use a system after they have learnt it, and how pleasant a system is to use.

Usability is not a simple property of a user interface, but has multiple components [82]. It is commonly associated with learnability, efficiency, memorability, errors, and satisfaction. By defining the abstract concept "usability" in measurable components, it is possible to achieve an engineering discipline where usability is "*systematically approached, improved and evaluated (possibly measured)*" ([82], p. 26).

Usability can be engineered into a product through an iterative design and development process that is referred to as usability engineering ([83] cited in [80]) which begins with identifying users, analyzing tasks, and setting usability specifications, and continues with developing and testing prototypes, through iterative development and test cycles.

3.1 Usability standards

Since the company involved in developing the evaluation framework in cooperation with U-ODD strived to base their usability testing on the industrially accepted ISO standards, it is worthwhile examining the contents of these standards at some depth.

One function of standards is to impose consistency, and although attempts have been made to do this through the use of the ISO standards, in many areas industry standards have been more influential and the ISO standards have not been widely adopted [84]. However, one ISO standard which has had an impact [84] is ISO 9241-11:1998, Guidance on usability [26], concerning measures of usability. The approach to software in ISO 9241 is based on guidance and principles for design that permit design flexibility [84]. Some usability metrics have been defined by the software engineering community, and are included in ISO 9126-4 [85], which includes detailed metrics for quality in use, that expand on the measures of usability in ISO 9241-11 [84]. A third standard within this area is ISO 13407:1999, Human-centred design processes for interactive

systems [86], which describes the essential user-centred design activities that are needed to produce usable products [84].

According to ISO 9241-11:1998, Usability is 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. Effectiveness is the accuracy and completeness with which users achieve specified goals. Efficiency concerns the resources expended in relation to the accuracy and completeness with which users achieve goals. Efficiency in the context of usability is related to 'productivity' rather than to its meaning in the context of software efficiency. Satisfaction concerns freedom from discomfort, and positive attitudes towards the use of the product.

ISO 9241-11:1998 states that it is necessary to identify goals and decompose effectiveness and satisfaction and components of the use context into sub-components with measurable and verifiable attributes when specifying or measuring usability. For an illustration of the components and the relationships between them, see Figure 3.1 [26]. According to the ISO standard, when measuring or specifying usability it is necessary to describe the intended goals, to describe the components of the context of use, including users, tasks, equipment and environments, and to specify values for effectiveness, efficiency and satisfaction for the intended contexts [ibid].

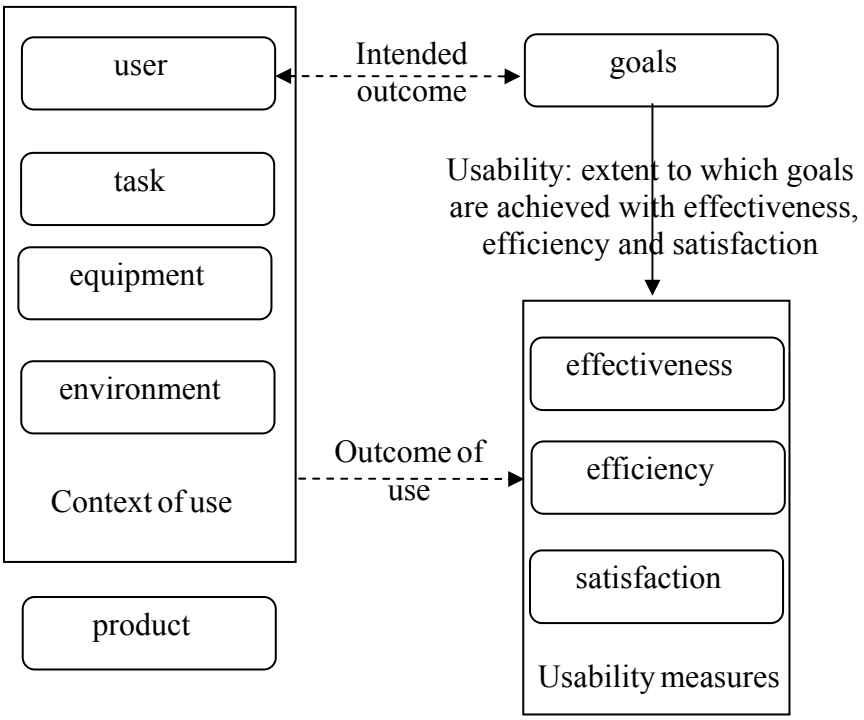


Figure 3.1. The usability framework.

The following relevant terms are also defined in ISO 9241-11. Context of use deals with the users, tasks, equipment (hardware, software and material) and the

physical and social environments in which a product are used, Goals are intended outcomes and Tasks are the activities required to achieve a goal. A User is a person who interacts with the product.

ISO 13407:1999 "Human-centred design processes for interactive systems" is an international standard concerning usability and user centred design, which also makes use of the above mentioned definitions concerning usability as defined in ISO 9241:11 [20].

ISO 13407:1999 "*provides guidance on human-centred design activities throughout the life cycle of computer-based interactive systems. It is aimed at those managing design processes and provides guidance on sources of information and standards relevant to the human-centred approach*" ([86] p. 1). It defines human centred design (HCD) as "*an approach to interactive system development that focuses specifically on making systems usable. It is a multi-disciplinary activity which incorporates human factors and ergonomics knowledge and techniques*" ([86] p. iv).

HCD has four main principles and four main activities (see Figure 3.2 for an illustration of the activities), which are closely connected.

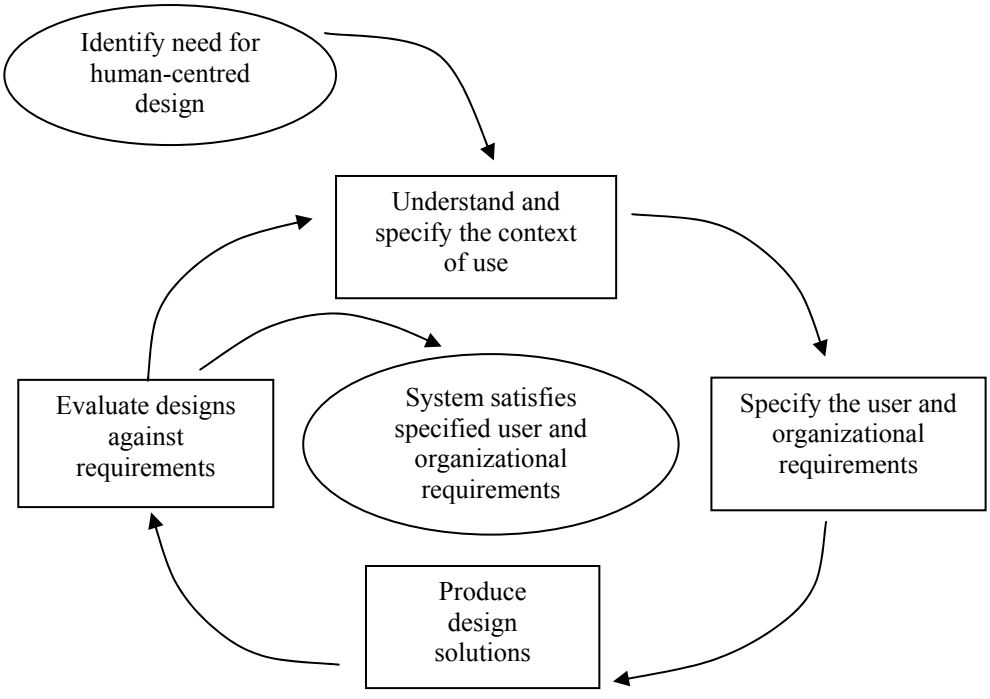


Figure 3.2. Human Centred Design [86]

The first principle is active involvement of users, and a clear understanding of user and task requirements. This user involvement provides valuable knowledge about the use context and how users are likely to work with a future system or product. Even when developing generic or consumer products, which by their nature have a dispersed user population, it is still important to involve users or "*appropriate representatives*" in development, in order to identify relevant

requirements and to provide feedback through testing of proposed design solutions.

The first activity is therefore understanding and specifying the use context. The context is to be identified in terms of the characteristics of the intended users, the tasks the users are to perform and the environment in which the users are to use the system. In the context where we have worked, this is not a trivial task. As previously mentioned, we are working in rapidly changing mass market situation, with a product that is aimed at satisfying a number of market segments simultaneously. This situation demands a great deal of knowledge of end users and their current and expected use of mobile phones, in a situation that can change over a short period of time if, for example, a competitor releases a new model with new and exciting features. Fortunately, there was a great deal of branch knowledge within the company, and within the departments within the different companies that the company worked together with. This meant that the people who were involved had their finger on the pulse of the market, and were aware of what was happening in the marketplace, even though details of coming models are shrouded in secrecy. There is a great deal of information to be gleaned from many sources, and this information together with branch knowledge meant that it was possible to understand and specify the use context.

The second principle is an appropriate allocation of function between users and technology, which means the specification of which functions should be carried out by the users and which by the technology.

The second activity is therefore specifying the user and organisational requirements. A major step in most design processes is specifying functional and other requirements for products or systems, but in HCD this must be extended to create statements of user and organizational requirements in relation to the use context. This includes identifying relevant users, formulating clear design goals, and providing measurable criteria for testing the emergent design, that can be confirmed by users or their representatives. Once again, in a mass market situation, where the product is aimed at many potential customers, this is a potentially difficult task. In our study, the use context, and therefore the requirements, is dependent on the type of user and her or his use of the phone. Once again, there was an extensive knowledge base regarding phone use, much of which was gained through having access to panels of users, and people who were recurrent phone testers.

The third principle is the iteration of design solutions. Iterative design allows the results of testing “real world” design solutions to be fed into the design process. Combining user involvement with iterative design minimizes risks that systems do not meet requirements, and feedback from users is an important source of information.

The third activity is therefore to produce design solutions. Potential solutions are produced by drawing on the state of the art, the knowledge of the participants, and the context of use. The process involves using existing knowledge to develop design inputs with multi-disciplinary input, concretizing

the design through simulations, mock-ups, etc., presenting the design to users and allowing them to perform tasks – real or simulated – and altering the design in response to user feedback. This process should be iterated until design objectives are met. The interaction design department at UIQ has been a good example of the type of actor involved in this process. They had a mix of many different competencies, from graphical designers, interaction designers, and usability experts, working closely not only with customers and end users, but also together with development teams within their own organisation. This method of working allows the design and development of features that are acceptable in the marketplace, and are feasible to develop.

The fourth principle regards multi-disciplinary design, and states that HCD requires a range of personnel with a variety of skills in order to address human aspects of design. The multi-disciplinary design teams need not be large, but should be capable of making design trade-off decisions, and should reflect the relationship between the customer and the developing organisation.

The fourth activity is to evaluate the design against requirements. Evaluation is an essential step in HCD, which can be used to provide feedback that can improve design, can assess whether design objectives have been met, and can monitor long term use of the system. It should take place at all stages of the system life cycle [86]. This is where the usability evaluation framework is the central factor in our study, and in this thesis, allowing the evaluation and validation of solutions, from a feature level to the level of the complete product.

The four principles have led the work that has led to the evaluation framework on two levels, both in the setting of the research cooperation and in the industrial setting. Active involvement of users has meant that the participants in the research cooperation have been the people who are most involved in the daily work of testing, and those who are most in need of their results. It has also meant that the evaluation framework has been designed so that the end users in the marketplace have been central figures in the testing process.

In the design of the framework, the primary function has been the observation of use case performance, and the lessons that can be learned from this, so the allocation of function between users and technology has meant that the test leader has been supported by basic technology for taking notes, registering use case performance time, and noting events that have caught his or her attention, and noting testers' comments. As the test has developed, the need for more complex tools has arisen, leading to the development of computer based tools for registering test data. In the testing itself, there is a natural division between the users and the technology, since the tests are performed on a mobile phone, and the actions performed by the user are bounded by the technology.

Regarding the iterative process, this has once again been apparent on the two levels, where the design of the evaluation framework has been an iterative process where studies of the work of the testing staff has led to developments in the evaluation framework, and studies of the testing process together with end users have also led to development of the framework. The final principle, calling for multi-disciplinary teams, is also apparent on these two levels, where

the researchers have worked together with a number of industrial participants with differing skills and functions within the organisation, and where the work of testing and the analysis of results has been the responsibility of a multi-disciplinary team that has had the freedom to develop the framework in the way they have best seen fit.

3.2 Usability testing

Nielsen and Mack distinguish between empirical evaluations, where real users test the interface in order to assess usability, and usability inspection, which is a generic name for a set of methods where evaluators, who can be usability experts, end users, or other types of professionals, examine the usability-related aspects of a user interface. The defining characteristic of usability inspection is that it is relies on the judgement of the inspectors, although the criteria that the judgement is based upon vary between the different individual methods [81].

Usability inspection is a term that is coined in reference to inspection methods that have long been common in SE for improving code. Usability inspections, in the form of e.g. Cognitive Walkthroughs (see e.g. [87]) is an area that has received attention within the SE field. However, although they have their uses, usability inspection methods cannot however replace empirical testing, as they fall short of empirical methods, and laboratory and field data are richer and more representative of real users of a proposed system [88].

According to Nielsen, usability testing is the most fundamental usability method, which is in some sense irreplaceable [82]. This is so since it provides direct information regarding people's use of computers, and of their problems with the interfaces that are tested. The primary characteristic of a usability test is that its goal is to improve the usability of a product, through a test that is planned on the basis of specific goals and concerns. Other characteristics are that the test is performed by real users who perform real tasks, that the test is observed and recorded, and that the data obtained is analyzed to diagnose problems and to recommend measures to fix the problems [80].

Usability testing is an empirical method that can be in the form of laboratory testing or field testing. A recent study has shown that laboratory testing, whilst attempting to simulate everyday usage, cannot account for uncontrollable factors that influence mobile phone usage in everyday life [89]. More types of and occurrences of usability problems were found in the field, and these tended to be more critical regarding usability. It has also been found that although usability testing in the field is time consuming, and complicated, compared to lab testing, it is worth the extra effort as it identifies more, and more serious, usability problems than lab testing [90].

Usability tests can be useful in a competitive marketplace. They can be performed on competitor's products that are already on the market, either in isolation or in comparison to a version of your own product, in order to understand their strengths and weaknesses, and to help your product meet users' needs better than the competitor does. They can also be used to make design decisions, by testing different proposed design solutions [80].

In relation to the above, the evaluation framework is an example of usability testing, rather than usability inspection, relying as it does on an end user testing the device by performing core tasks in a context of use. The use context is central in the process, concentrating as we do on capturing the members' point of view, and the best way to capture this is by testing in as realistic a setting as possible. The data from testing is recorded and analysed, and via a feedback process that can be adapted to the needs of different stakeholders is used to recommend measures to solve problems.

The evaluation framework is a hybrid method. It is not a pure field test that involves testing in, for example, an outdoor environment with different types of interference from the outside world, but neither is it a lab test where the tester is isolated from the real world. The test takes place in an everyday environment where the tester feels comfortable, and unless the intention is to test a particular feature that may not be amongst the most common features, the test is performed using everyday use cases that the tester feels comfortable with.

The test package is designed for use in the competitive marketplace where the company is working, and as stipulated above, is designed to be used for testing in a range of situations, from features and proposals that are at the design and prototype stage, or comparing different models and products, to testing devices preparatory to release on the market.

Dumas and Reddish found that the character of usability testing had changed between 1993 and 1999 [80]. Usability testing has become more informal, more iterative, and more integrated. Specialists more often remain in the room with the participants and more active intervention takes place, which entails the tester sitting in the room actively probing the participant's understanding of what is being tested. Many tests rely on qualitative rather that quantitative data, meaning that they concentrate more on identifying problems and less on proving those problems by means of measurement of time and errors. Reporting has also become less formal. Amongst reasons for these changes is the fact that there is a need for results that can be used in rapid development and release cycles, but also that there is a maturity within and acceptance of the field of usability, where usability specialists no longer need to justify the methodology or results of a test.

This shift towards informal, iterative and integrated testing was very apparent in the industrial context where this research cooperation has taken place. The structure of the evaluation framework necessitates the presence of the test leader, since it is this key person who is a bearer of the knowledge that can be gathered from observing the test and analysing the test data. Even though the test leader in our case does not usually actively probe the tester's understanding of what is being tested, she or he does have the opportunity to ask follow up questions if anything untoward occurs, and is able to converse with the tester to glean information about what has occurred during the test.

The evaluation framework is dependent on qualitative data, but in our case, in contrast to what is stated above, it is also dependent on quantitative data, and the qualitative and quantitative data are used to support one another when

analysing or presenting the findings. We found that there were many occasions when the test results could be presented in informal ways, and the framework was useful for quickly getting results back to the development organisation, for use in the rapid development cycles that are mentioned above, where the methodology is accepted, and competence of the test leader is taken as given. However, we have also found that there are situations where some stakeholders are dependent on a more formal presentation of results. For a discussion of this, see Chapters 5 and 6, where we present the results of two case studies on this theme.

Since usability testing is seen as a valuable method for diagnosing problems rather than validating products, a smaller number of test participants is required in order to achieve valid results. This is also dependent on the fact that usability testing has become more iterative, and it has been found that very small samples, typically 3 to 6 people per testing round, can be used to evaluate products, as testers are involved earlier and more often in the process [80]. Dumas and Reddish [80] refer to previous studies that indicate in one case that almost half of all major usability problems were found with as few as three participants, and in a second case that a test with four to five participants detected 80% of usability problems, whilst ten participants detected 90% of all problems. This indicates that the inclusion of additional participants is less and less likely to contribute new information. The number of people to include in a test thus depends on how many user groups are needed to satisfy the test goals, the time and money allocated for the test, and the importance of being able to calculate statistical significance [80]. If a whole product line depends on the test object, then it may be necessary to measure to a high degree of certainty, but in most cases this is not necessary, and a typical usability test includes six to twelve participants in two or three user groups.

In our study, we have also found that in general, few participants are needed in a test series in order to gain indications of usability problems. In discussions, the test leader estimates that problems are usually apparent after three to five tests. As in the above, however, tests have been performed with as many as 48 participants, in order to gain a cross section of the population of phone users.

3.3 Usability metrics

Metrics are an important factor in development processes, and measurable usability requirements mean that usability work in projects becomes more goal-driven and finds greater acceptance. Measurability is an important factor in development cultures where '*what is measured gets to be done*' [91].

Usability engineering requires the setting of specific quantitative usability goals. There is a general agreement from standards boards regarding the fact that the dimensions of usability are effectiveness, efficiency and satisfaction, and also, though to a lesser degree, which metrics are most commonly used to quantify these dimensions [92]. There are however problems regarding the use of usability metrics, which are measured on their own scales and are clumsy and difficult to use. In order increase the strategic use of usability data, it is

necessary to represent the usability construct in a precise fashion. There are existing methods such as a number of questionnaires that are often used by practitioners as a way of expressing usability with one number, but expressing the whole construct of usability on the basis of these is questionable. The existence of this practice does however illustrate the fact that there is a need for metrics to represent usability in a clear and manageable form [92].

In a rapidly changing context, it is difficult to find stable values that can represent usability goals, since the preconditions can change from day to day, and the figures that are found to be acceptable one day may be out of date after a short period of time. Therefore, we have put a lot of effort into this area to develop ways of illustrating the usability construct, without being dependent on predefined usability goals. Even though we use questionnaires to collect usability data, and can calculate metrics on the basis of these, these metrics are never used in isolation, and are never presented as numbers. They are always contrasted with other data and presented visually, and can be compared to previous results, even though the values that the metrics represent may have changed in magnitude. For more information on how the results can be visualised, see Chapter 4 and Appendix A. See also Chapter 6 for a discussion of how the results can best be presented.

One area that can be experienced as a problem is setting reasonable criteria for performance measures. This can be especially difficult regarding setting time limits, if there is no user data on which to base the measures. This highlights the importance of experience when setting goals, and the importance of watching users perform tasks, in order to gain an impression of what is reasonable [80]. As mentioned previously, we have not concentrated on setting goals, since the context where we work is so changeable. In our evaluation framework we have progressed beyond setting time limits for measuring performance. This was done in the beginning of the process, and records were kept of the time taken in order to make comparisons with previous results. Time taken is one indicator of performance, but in a rapidly changing environment, what is an acceptable time can change rapidly, and even though time taken can be used to measure efficiency, taken by itself it tells us little about how satisfied the user is with the device. Instead, we use the spread of times taken for all users to complete a given use case and contrast this with other measures, and we observe what happens during the period that the tester performs the use case, in order to capture other equally useful data.

Measuring usability involves the collection of both performance measures, and subjective measures [80]. Performance measures are quantitative counts of visible actions. It is possible to measure factors such as time taken, and errors in operation. For most observable behaviours, these measures do not require any element of judgement. Subjective measures are people's perceptions, and judgements. They can be either qualitative or quantitative. Both objective and subjective measures are taken during the testing. The objective measures are such things as time taken to perform a use case, and the subjective measures are user attitudes to the device and the use case performance, but also the test leader's judgement on how well the device performed. These are complemented

by the information that is gathered by the test leader, which is sometimes possible to register in a spreadsheet, in the form of comments made by the tester, but is often more subtle, and is stored instead as thoughts and reflections in the mind of the test leader. For more details about this, see Chapter 4.

The above is a very brief overview of the field of usability, testing and usability metrics, and how we work with these factors, and this serves as a background to the next section, which is a look at quality in the context of usability.

3.4 Quality, usability, and metrics

For general purpose products, such as the mobile phones that we have been concerned with in our study, the fact that products can only have quality in relation to their intended purpose is a key reason for being concerned with user-perceived quality [93]. But the mobile phone industry as a branch focuses on providing the market with new and improved technology rather than fulfilling end-user needs. It is a challenge to develop a usability test for a mass-market product, where specific end-user groups must be targeted [3], whilst being unwilling to exclude other potential end-users [4]. The evaluation framework that we have developed is a way to meet that challenge.

Bevan [93] states that the objective of software development is quality of use, and that software product quality is the way to achieve it. Many aspects of software quality contribute to quality of use, but ease of use is often critical for interactive software. Therefore we will now look at some of the connections between quality and usability, and metrics for illustrating them.

Quality is a complex concept, meaning different things to different people, and in order for others to understand your view of quality, it must be defined in a measurable way [94]. There are however many reasons why it is unclear what a good measure of software quality is. Wong and Jeffery, for example, found that developers and users have different cognitive models of what software quality is, with differences in their choice of characteristics and consequences and in their desired values [95].

Therefore, there is a variety of interpretations of the meaning of software quality, the meaning of the terms to describe its aspects, of which aspects should be included in a model of software, and which software development procedures should be included in the definition. Recent research has adopted the user-based view of quality, recognising that each person is different and has a different perception of quality [95].

Kitchenham and Pfleeger state that the user view is concrete, and is grounded in product characteristics that meet the user's needs. Users are concerned with aspects of usability, not only with reliability. Studying usability is also related to the user view, as researchers observe how users interact with products [94]. This is also related to the standpoint that we have taken regarding testing, where we observe the interaction that takes place with the device. The testing is concerned with capturing and measuring the user's subjective view of the device, rather than testing to isolate failures in the software, or testing reliability, and it is a complement to the testing methods that achieve this.

It is apparent that there are many meeting points between the user view of quality and usability. However, confusion even exists as to the meaning of the term usability. This is one of the reasons for problems incorporating usability engineering into the field of software engineering (see for example [93], [36]). Bevan distinguishes two separate approaches to usability, reflecting the problems that are apparent in divergent ideas regarding quality. One is a product-oriented view, which sees usability as ease of use, whilst the other is a "top-down" approach, where usability is seen as the ability to use a product for its intended purpose. In the product-oriented view, which fits well in conventional software engineering, usability is seen as an independent contribution to software quality, whilst from a human factors viewpoint, usability can only be achieved through a process of user-centred design [93]. This is the approach that we have taken in our study, designing the test from a human centred design standpoint, and ensuring that the evaluation framework also complies with this standpoint. This is also one of the areas where we contribute to knowledge in the field of software engineering, having developed, used and evaluated the evaluation framework in a software engineering environment.

Metrics are an important factor in finding acceptance for studies of quality and of usability. When incorporating usability in a quality system, measurable usability requirements need to be specified, and the fulfilment of goals must be monitored during design [93]. Metrics are important in development processes, and measuring usability requirements leads to greater acceptance for usability work [91]. In a usability test, it is important to measure all three aspects of usability: satisfaction, efficiency and effectiveness rather than simply assuming that they are correlated [96]. This is one of the approaches that we have taken in our study: finding metrics that are easy to collect in the environment where we perform the testing and that are acceptable to the stakeholders who are dependent on the testing results. However, as previously mentioned, we no longer set specific goals in the form of absolute metrics to be achieved. Not all stakeholders, however, require the results to be directly connected to the metrics that are collected, and sometimes the knowledge of the test leader gives sufficient information. For other stakeholders it is important that the results that are presented visually are directly accountable and can be led back to the data.

To briefly summarise this chapter, it has shown how usability is discussed and treated in several fields, and in which way the work we have performed is grounded in these discussions, theories and practice. Although much of the material is taken from outside the field of software engineering, it is still clear that usability and quality are important factors in the field, and much work is being done on the subject. The evaluation framework and this thesis are further contributions to that work.

Quality and usability are intertwined. According to Bevan [93], quality of use is the extent to which a product satisfies stated and implicit needs when used under stated conditions. Quality of use is expressed in the ISO standards, which are central to the work performed in this study and within the company, as the extent to which specified goals can be achieved with effectiveness, efficiency

and satisfaction by specified users carrying out specified tasks in the specified environment [26]. This places the focus of quality on measurable aspects of the product, the tasks and the context in which it is used, rather than on the product in isolation. This is a worthwhile approach but is still too narrow as it does not cover enough of the quality aspects related to the users' experience.

As shown above, and in the rest of this thesis, the evaluation framework is an attempt to remedy the above shortcoming, whilst still retaining the broader and measurable aspects of usability. It collects metrics that measure the elements of usability, but also collects metrics and knowledge of the user experience [97]. In doing so, it gives a clear demonstration of quality, from the customer and end-user point of view.

In the next chapter, we present an overview of the evaluation framework, the background to its development, the philosophy behind it, the research process that has led to it, and some of the mechanics of the testing process itself.


usability
results
test
chapter
four
user
process
development
UTUM
use
attributes
testing
Metrics
satisfaction
product
customer
experience
based
information
research
developing
line
Must-be
basis
industrial
tool
new
improvements
provide
rather
giving
requirements
SUS
corresponds
made
observations
scale
functionality
find
designers
leader's
One-dimensional
needs
included
help
functional
value
Kano
way
Thus
may
questionnaire
whilst
together
input
see
step
field
phone
Figure
diagram
discussions
software
mobile
present
innovative
presenting
interaction
cooperation towards
measuring
performed
Show performance
observed
general
cases
case
already
first
within
customers
must
UIQ
solve
part
since
market
right
able
level
improvement
well
Given
engineering
quantitative
phones
presented
method
methods
thereby
long
feature
specific
good
model
expert
ways
Technology
future
concerns
one
two
used
company
users
design
important
role
stakeholders
illustrates
time
attempt
quality
metric
version
even
point
framework
understanding
order
team
different
although
also
qualitative
second
developed
perspective
leader
found
practice
Attractive
effectiveness
study
Symbian
characteristic
problem

Chapter Four

4. UTUM – a description and explanation

This chapter is an introduction to the UTUM usability evaluation framework. UTUM (UIQ Technology Usability Metrics) is an industrial application that has been developed and evolved through long term cooperation between an academic partner, BTH/U-ODD, and UIQ Technology, an industrial partner with long experience within the mobile phone branch. UTUM is grounded in usability theory and guidelines, and in industrial software engineering practice and experience. It is in use within an industrial setting heavily influenced by software engineering, and has been the subject of continual joint development, at the same time as it has been the subject of academic study.

UTUM is an attempt to bridge a gap between the software engineering and the HCI communities. The purpose of developing the framework was to find a method that is based in usability practice and theory, capable of measuring and presenting usability in a way that can be accepted by the industry at large and in the field of software engineering. In this chapter, we describe and explain the evaluation framework, and some of the principles behind it.

4.1 What the test aims to solve

In the industrial environment where the company has been active, changes are rapid. Given the fact that usability and user experience are such important factors when buying and owning a mobile phone, it is vital for a company to measure and visualise the usability and user experience offered by the devices that they design, develop for and market, together with different mobile phone manufacturers. The method must give results quickly, and it must be sensitive to changes in the competition situation within the marketplace. It must also be reliable and accountable. The results of the test must be presented to a variety of stakeholders with varying information needs, and the results shown should be able to be traced back to the metrics that they are based on.

To help solve this, UTUM was developed. UTUM is "a method to generate a scale of measurement of the usability of our products on a general level, as well as on a functional level" [98]. As an indication of the importance given by UIQ Technology to UTUM, it was presented to the telecom industry at the Symbian Smart Phone Show in London, October 2006 [99]. A more detailed explanation of the test is available on UIQ Technology's website [100]. The evaluation framework is presented in even greater detail in a white paper that can be downloaded from UIQ's website, entitled The UTUM report: Inside Information [101]. A brief video demonstration of the whole test process (approximately 6 minutes) can also be found on YouTube [102]. The purpose of UTUM is to be a cost effective tool for guiding design decisions that involves users in the testing process.

4.2 The need for a philosophy

In the white paper, The UTUM Report [101], UTUM is presented as being innovative in software development practice in two ways. Firstly, through the relationship to users and the attitude towards their participation in the design process: in the same way as the research process, the test itself is inspired by PD and an ethnographic approach to testing, in order to gain a deep understanding of how users understand the use of mobile phones. Secondly, it is innovative in how the understanding gained in the test process is directly incorporated into the development process, through the central role of the test expert who is a part of the design team, and through the use of clear and simple metrics that open a path to understanding the statistics that are a basis of the results.

There are several distinctive characteristics of UTUM. The first two of these are concerned with the relationship with users and how user input is received and perceived, whilst the other two deal with software development practice and concern organisation and method development within the company.

The first characteristic is the approach to understanding users and getting user input. Instead of simply observing use, the test expert interacts and works together with the users to gain insight into how they experience being a mobile phone user, in order to gain an understanding of the users' perspective. Because of this, the users that help with the testing are referred to as testers, because they are doing the testing, rather than being tested. The representative of the development company is referred to as the test leader, or test expert, emphasising the qualified role that this person assumes.

The second characteristic deals with making use of the inventiveness of phone users. This entails giving the users a chance to actively participate in the design process. The participatory design tradition respects the expertise and skills of the users, and this, combined with the inventiveness observed when users use their phones in real life situations, means that users provide important input for system development. The test expert once again has an important role to play as an advocate and representative of the user perspective. Thus, the participation of the user provides designers, with the test expert as an intermediary, with good user input throughout the development process.

The third characteristic is the continuous and direct use of user input in design and decision processes. The high tempo of software development in the area of mobile phones makes it difficult to channel meaningful testing results to the right recipient at the right time in the design process. In an attempt to alleviate this problem, the role of the test expert has been integrated into the daily design process. The results of testing that is performed in-house can be channelled to the most critical issues, and the continual process of testing and informal relaying of testing results to designers leads to a short time span between discovering a problem and implementing a solution.

The fourth characteristic concerns presenting a summary of testing results in a clear and concise fashion that still retains a focus on understanding the user perspective, rather than simply observing and measuring user behaviour. The results of what is actually qualitative research are summarised by using

quantitative methods, giving decision makers results in the type of presentations they are used to dealing with. The statistical results are not based on methods that supplant the qualitative methods that are based on PD and ethnography, but are ways of capturing in numbers the users' attitudes towards the product they are testing.

Although these four characteristics are highlighted separately above, in practice they are not four methods used side by side, but are part of one formal method that works as a whole. An advantage of using this type of formal testing method, even though many of the results are communicated informally, is traceability. It is possible to see whether improvements have been made, and even trace the improvements to specific tests that have been performed, and thereby make visible the role of the user testing in the design and quality assurance process.

4.3 Early versions of the test

The process of developing the usability evaluation framework began in the period between 2001 and 2004, with an attempt, initiated by Kari Rönkkö, to introduce Personas in the development process. This was done in order to bridge a gap between designers and developers, and other stakeholders in the company. The attempt was abandoned when Personas was found to be unsuitable for the company, for reasons that can be found in [4].

At the same time, in 2001, Symbian company goals included the study of metrics in the development process, and an early version of the user test was developed by Patrick W. Jordan, then head of the interaction design group at Symbian, Mats Hellman, from the Product Planning User Experience Team at UIQ Technology, and Kari Rönkkö. The test was implemented as a two part tool, consisting of six use cases to be performed on a mock up [6], and the use of the System Usability Scale (SUS) [103]. The test was performed at several places in the development process, in order to show improvements in usability. The results were seen as somewhat predictable and did not contribute greatly to the development process in practice, although they did show that the test method could lead to a value for usability.

In the period between 2004 and 2005, a student project, supported by representatives from U-ODD, studied how UIQ measured usability, and pointed out improvements that could be made, such as the inclusion of user investigation, a way of prioritising use cases, and being able to include the results of the test leader's observations in the method.

The first UTUM version was developed in 2005, consisting of three steps. The first step was a questionnaire used to prioritize use cases, and to collect information for future analysis and statistics. The use cases could be decided in advance, if the company had specific features that were prioritised for evaluation, or decided on the basis of the answers to the questionnaire. Each use case was observed and timed by the test leader, and videotaped if this was considered to be necessary. After every use case, the user filled in a small user satisfaction evaluation, explaining how well the device supported their

intentions and expectations for that specific use case. The second step was to calculate a performance metric, based on completion time for the specific use case, resulting in a value between 0 and 1. The third step was an attitudinal metric based on the SUS, also resulting in a value between 0 and 1. These results were used to calculate a Total Usability Metric with a value between 1 and 100. There was still no way of including the test leader's observations in the metrics, but besides these summative results, the test leader, based on observation of the test process, could also directly feed back formative testing results to designers, giving user feedback to help the improvement and redesign work.

In 2005 a usability engineer was engaged in developing the test, and the author of this thesis became involved in the process and began to study the field of usability, observed the testing process, and interviewed staff at UIQ. In an iterative process during 2005 and 2006, a new version of the test was produced, UTUM 2.0, which included more metrics, and new ways of interpreting and calculating the metrics.

As can be seen, the test has a long history of development, and UIQ and BTH have both had central roles in the process. In the following, we present some of the ideas that we have contributed to the process.

4.4 Taking inspiration from Kano for the presentation of metrics

In discussions during late 2005, we introduced the concept of the Kano model (see e.g. [15], [104]) into discussions together with the participants who were involved in developing the UTUM test.

The Kano model is a tool to gain an understanding of customer satisfaction (see Figure 4.1). It divides product attributes into 3 categories, which affect customer satisfaction in different ways. These are: One-dimensional attributes; Must-be attributes, and; Attractive attributes [104].

One-dimensional attributes: In the figure, the vertical line in the diagram indicates how satisfied the customer is, whilst the horizontal line indicates how functional some aspect of the product is. Traditional ideas about product quality have assumed that customer satisfaction is proportional to product functionality, and to the level of fulfilment. The diagonal line in the diagram illustrates this. Good functionality results in customer satisfaction, and poor functionality results in dissatisfaction. The better the attributes are, the more the customer likes them. In the case of a car, for example, fuel consumption can be assumed to be a One-dimensional product requirement. Also known as "Spoken qualities", these attributes are usually explicitly demanded by customers.

Must-be attributes. The curved line at the bottom of the diagram illustrates the Must-be requirements. Customers take these for granted, and the presence of them leaves the customer neutral and does not increase satisfaction. However, the lack of the feature leaves the customer dissatisfied. For example, having good brakes on a car does not raise the level of customer satisfaction, whilst having poor brakes causes dissatisfaction.

Attractive attributes: The curved line at the top of the diagram illustrates the Attractive attributes. If they are present, they delight customers, and they have the greatest impact on customer satisfaction. Fulfilling them leads to more than proportional satisfaction, but absence of them does not cause dissatisfaction, because they are not expected by customers, who may previously have been unaware of them.

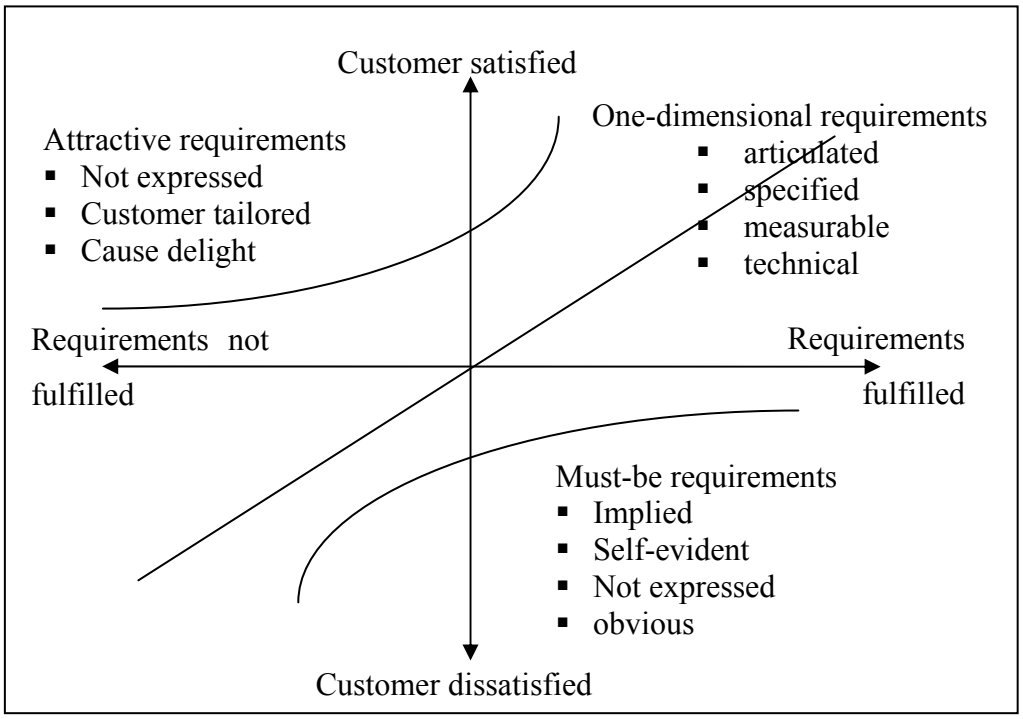


Figure 4.1. The Kano diagram

Thus, better performance will lead to more satisfaction, but not all satisfaction attributes are equal. The attractive attributes result more easily in customer satisfaction than must-be attributes, and attractive attributes increase satisfaction progressively with the improvement of the product performance.

These attributes may change category over time: attractive attributes can become one-dimensional attributes to finally become Must-be attributes. Customer needs, and thereby product attributes, are dynamic rather than static. Thus, customer satisfaction is more than a one-level issue. In competitive environments, such as the one where we have been working, it may not be enough to merely satisfy customers' basic needs. One strategy for success is to delight customers and exceed their wishes. This may require the development of innovative features.

There is some similarity with the use of the SUS, since a Kano questionnaire also makes use of Likert style questions. For each product feature, a pair of questions is formulated, which can be answered in one of five ways. The first (functional) question concerns the reaction if the product has the feature, the second (dysfunctional) concerns his reaction if the product lacks the feature.

The "voice of the customer" is important, and is a description of the problem to be solved from the customer's viewpoint.

In discussions, the head of the interaction design team said that he would like to try and connect the usability metrics program that was already in operation with quality as it was illustrated through the use of the Kano model. The ideal area to end up in would be above the line showing one-dimensional requirements, and preferably above the line where attractive requirements are found (see Figure 4.2 for an illustration).

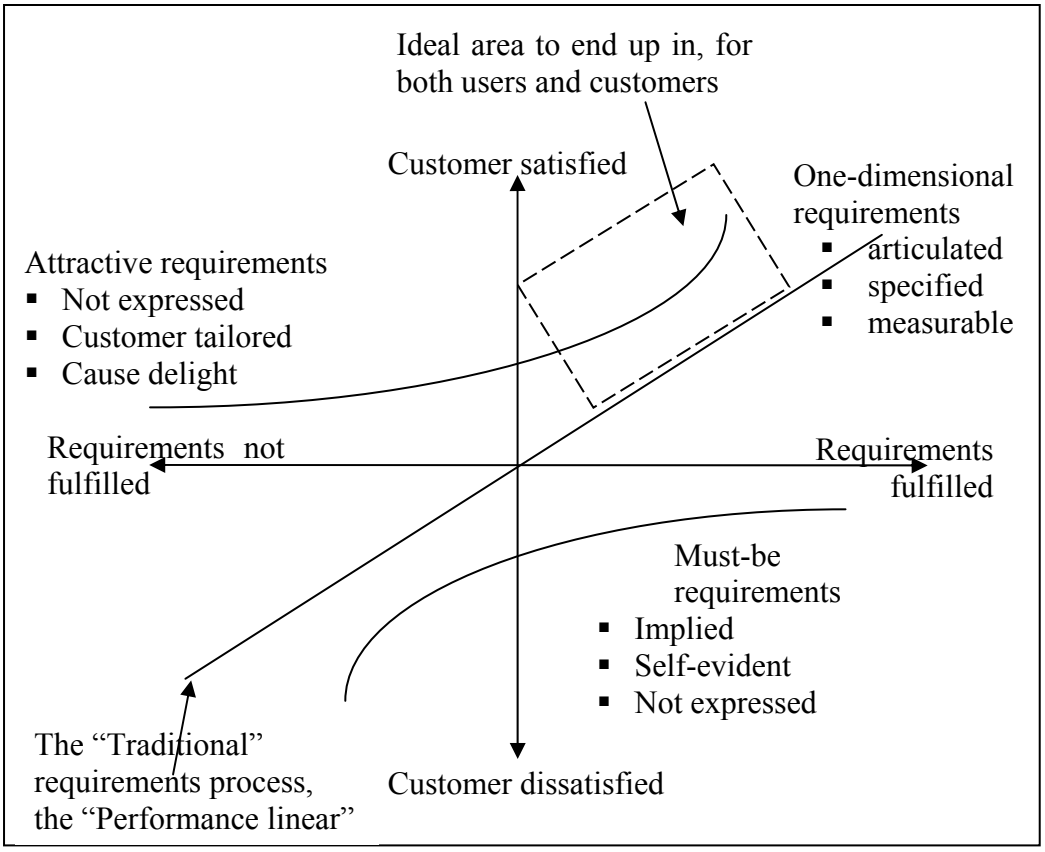


Figure 4.2. Discussions about Kano, image from research logbook

The discussions that were held regarding the Kano model led to the development of a way of presenting the metrics as plots in a 4-field diagram (see Figure 4.3). Since the metrics are calculated on the basis of a 5 point Likert scale, the point on the left or at the lowest point of the diagram corresponds to Strongly Disagree, the point on the far right or the top corresponds to Strongly Agree, whilst the intersection of the lines corresponds to No Opinion.

Figure 4.3. How results are presented [101]

The discussions about the Kano model also introduced the concept of "delighters", or the Wow factor, equivalent to the Attractive attributes found in the Kano model. These discussions also influenced the way of discussing and the thoughts about measuring user experience.

4.5 Summing up UTUM

To sum up UTUM, it is based on the same standpoint as the research cooperation that has led to it, with an acceptance of the importance of ethnography and participatory design. It has been developed in order to solve an important problem in the industrial context: that of measuring and displaying user attitudes regarding usability and user experience. It is a method for measuring and presenting usability, based on metrics for the traditional elements included in usability in accordance with the ISO standards: efficiency, effectiveness and satisfaction. This is combined with judgements made by a test leader/usability expert. The results of a UTUM test are stored as qualitative and quantitative data in spreadsheets, and as information in the mind of the test leader. The results can be presented to different stakeholders verbally, as diagrams and graphs, and in PowerPoint presentations.

The evaluation framework is a complement to other existing methods already used by the Interaction Design team and other departments within the company. As well as leading to new information about usability, it also supports the findings that they already arrive at. It shows another perspective and is another way of gaining acceptance for their results. It emphasises the quantitative aspects of the testing, thereby enabling comparisons.

It can identify areas where effectiveness can be improved, and can be used to measure the usability of specific phones or to compare different phones. Within the industry it has been found to be flexible and easy to use when developing new designs, and when evaluating implemented designs to find possibilities for improvement. It can find product strengths and weaknesses in order to direct development resources to the relevant areas. The results can be used to provide a common understanding of the usability of the product at a given time.

As stated above, the evaluation framework is suitable for the general user of a mass market product. But it also allows for a segmentation of users, and user groups, and allows the testing of specific functions on specific groups. The advantage of it is that it is a general method that can be easily used for a mass market product, but can also be adapted for use for specific purposes.

The direction that the testing is moving towards today is more and more concerned with User Experience, and in the future, theories and practice within this field are likely to affect the design and practice of the evaluation framework, and the presentation of results. For more information about User Experience and its connection to UTUM, see Chapter 7.

In the next chapter, we look at the results of the empirical studies that have been performed, and present the first of two case studies that have been performed as part of the research cooperation. This case study is an examination of the test package and how it stands in relation to the needs for formalised, metrics based results, and informal descriptions of usability problems.


chapter
Testing
five
test
Case
Study
Quality
Agile
Needs
software
framework
Formal
Product
research
process
results
development
usability
Organisational
agility
data
manifesto
work
Working
Owners
Designers
information
groups
different
customer
found
metrics
Figure
knowledge
customers
practice
important
based
material
leaders
value
processes
interest
qualitative
Assurance
balance
PO
plan
two
whole
spreadsheets
formalism
perspective
stakeholders
testers
must
items
demands
requirements
cooperation
following
changing
discussion
driven
need
industry
since
design
presented
Factors
interested
see
side
present
organizational
company
place
necessary
performing
result
still
effective
scale
easy
within
findings
change
large
analysis
regarding
performed
fact
help
methods
everyday
one
rather
right
tester
leader
effort
concern
aspects
shop
Even
also
relation
Individuals
successful
tests
collected
floor
related
organisations
comprehensive
obviously
documentation
situation
method
examine
management
al
good
particular
reasons
illustration
number
rapidly
seen
may
end-users
interaction
approach
roles
point
identify
changes
use
Improving
best
day
many
needed
points
However
period
already
mix
et
testers
methodology

Chapter Five

5. Meeting Organisational Needs – a Case Study

This chapter is a case study of the test package and how it stands in relation to the needs for formalised, metrics based results, and informal descriptions of usability problems. The chapter is based on the article: *Meeting Organisational Needs and Quality Assurance through Balancing Agile & Formal Usability Testing Results,* which was presented at CEE-SET 2008, Brno, is printed in the preprint of the conference proceedings [105], and is accepted for publication in LNCS.

The contribution it makes to the software engineering research community is that it is evidence of the need to combine agility and plan driven approaches in one usability evaluation framework.

It shows that 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 evaluation framework is seen as a successful hybrid method combining agility with formalism, satisfying organisational needs.

For a more detailed description of the methodology used in the case study, and a discussion of threats regarding reliability and validity, see Section 2.9.

5.1 Introduction

As mentioned in the introduction to this thesis, product quality is becoming a dominant success criterion in the software industry, and the challenge for research is to provide the industry with the means to deploy quality software, allowing companies to compete effectively [22]. Quality is multi-dimensional and impossible to show through one simple measure, so research should focus on identifying various dimensions of quality and measures appropriate for it, and a more effective collaboration between practitioners and researchers would be of great value. The criticality of software systems is another reason why quality is important (a view supported by Harrold in her roadmap for testing [23]) as are changes in legislation that make executives responsible for damages caused by faulty software.

The demand for continuous and rapid results in a world of continuously changing business decisions makes if impossible to rely on complete, testable and consistent requirements, traceability to design, code and test cases, and heavyweight documentation. This points to a need for agility. Both agility and quality are becoming more and more important [38]. The pace of change, due to changes in technology and related infrastructures, the dynamics of the marketplace and competition, and organisational change, demands an agile approach. 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 [24].

This chapter deals with one of our case studies of the usability evaluation framework. With the help of Martin et al. study [43] 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 produced in the evaluation framework are suitable for use in an agile process. Even though Extreme Programming 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 investigation in this case study is in reference to the work of Martin et al. [43] and 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 [43]). Both perspectives are needed!

The structure of the chapter 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 chapter ends with a discussion of the work, and conclusions.

5.2 Testing – prevailing models vs. agile testing

As quality becomes a dominant success factor for software, processes to support software quality will become increasingly important. Testing is such a process, performed to support quality assurance, and provide confidence in the quality of software. An emphasis on software quality requires improved testing methodologies that can be used by practitioners to test their software [23]. This short section therefore briefly discusses the field of testing, in connection with the fact that the evaluation framework is seen as an agile testing methodology.

The prevailing model of testing has been a phase based process, consisting of unit testing, integration testing, function testing, performance testing, acceptance testing, and finally an installation test. 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 [17].

Agile software development has changed how software development organisations work, especially regarding testing [41]. In for example Extreme Programming (XP) [42], testing is performed continuously by developers [43]. Both programmers and customers create tests, and some teams have dedicated testers, who help customers create tests [42]. The role of the tester is also changing, from a situation where the designer is the tester to a situation where these roles become separated, where testers are generalists with the kind of knowledge that users have, and where good testers may have traits that are in contrast to the traits that good developers need [44]. So there are 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.

5.3 The evaluation framework and the case

In the case in question, test usability leaders from two organisations in two different countries performed testing in parallel. Testing was performed in a situation where there are complex relationships between customers, clients, and end-users, and complexities of how and where results were to be used. This testing was performed partially to validate the evaluation framework itself as a method for quality assurance, but also to obtain a greater number of tests, to create a baseline for future validation of products, to identify and measure differences or similarities between two countries, and to identify issues with the most common use-cases. Normally, there is no need for such a large number of testers or data points (see Chapter 2 for a discussion about this in the field of usability). However, even though this can be seen as a large test from the point of view of the participating organisations, compared to their normal testing needs, with more than 10 000 data points, testing was still found to be a process where results were produced quickly and efficiently.

Since this is the first case study where the evaluation framework is presented, the following is a brief illustration of the evaluation framework and its contents. The information here can be complemented with the material in Chapter 4 and

Appendix A, which present the evaluation framework in greater detail.

Figure 5.1 is an illustration of 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. The stakeholders in the testing can be seen at the top of the flow, as interested parties. These stakeholders can be within the company, or licensees, or customers in other organisations, and their requirements influence the design and contents of the test.

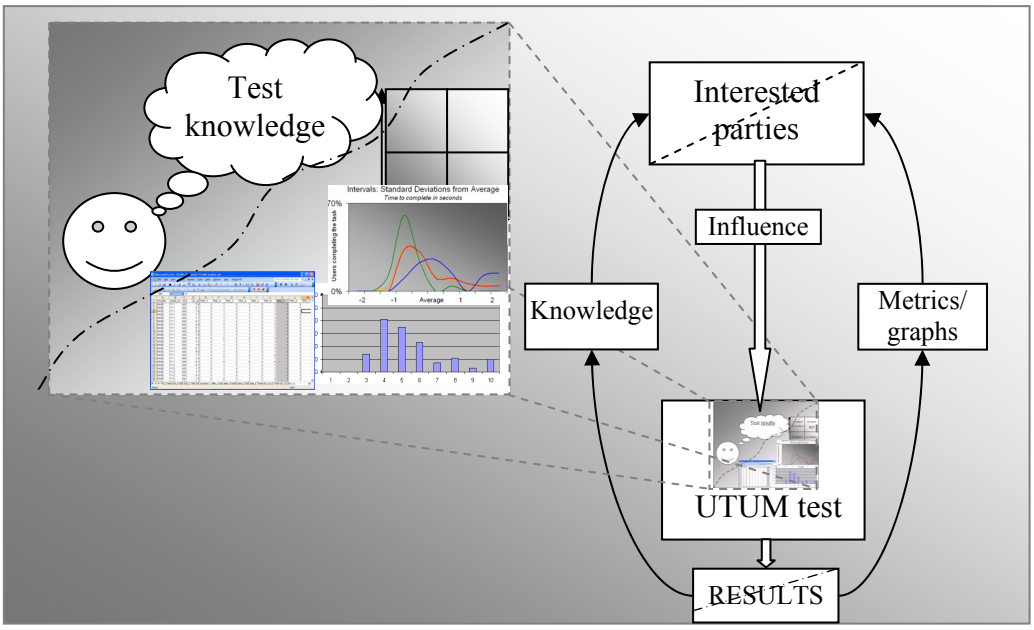
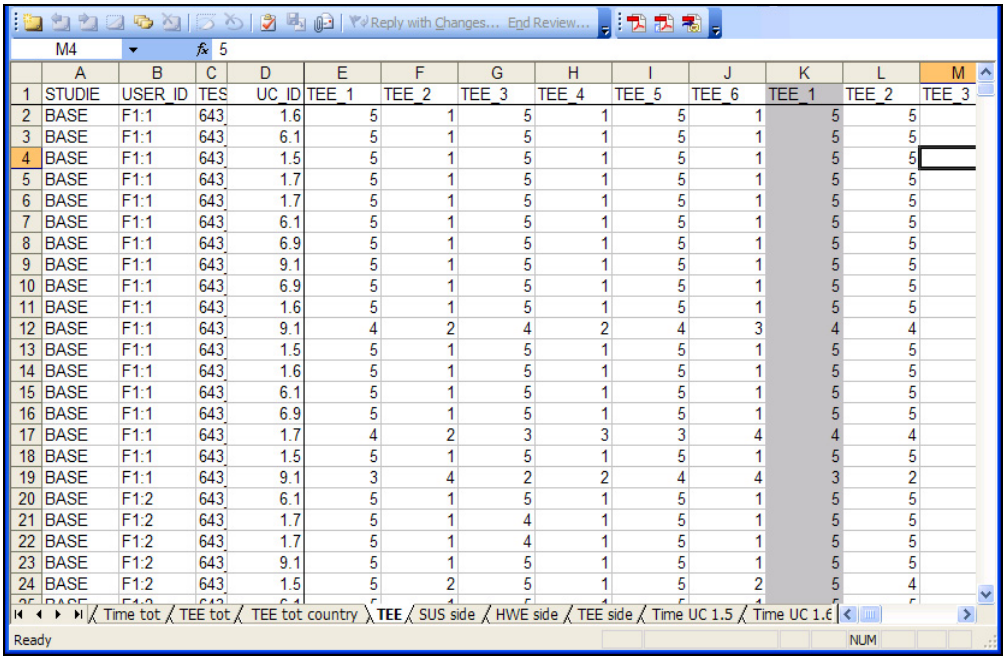


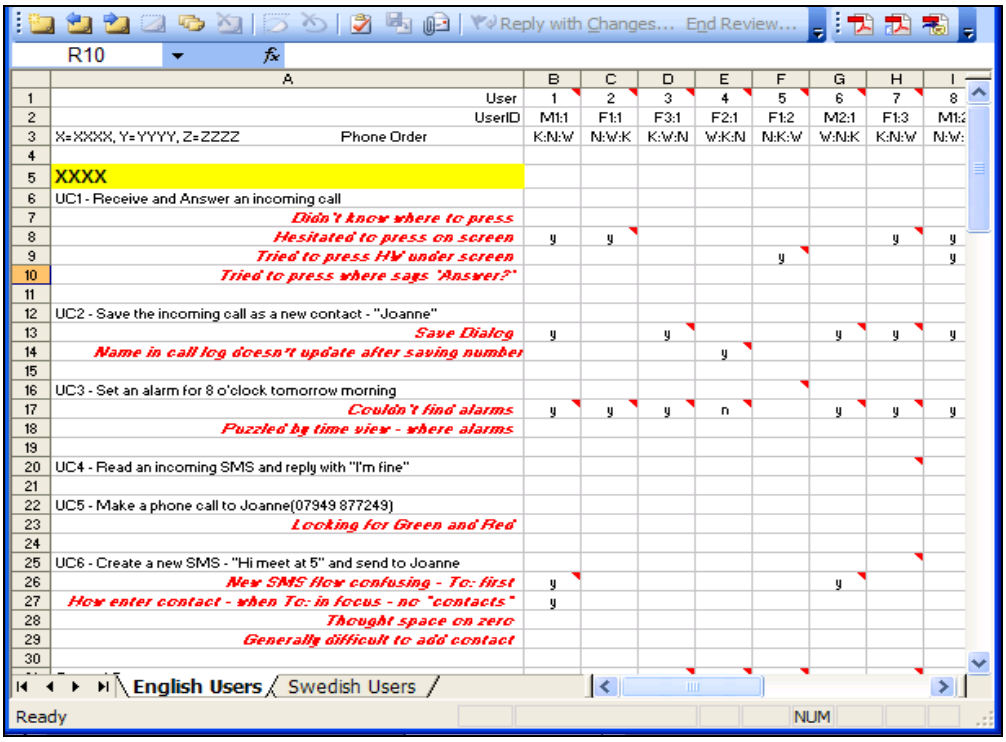
Figure 5.1. Contents of the evaluation framework, a mix of metrics and mental data

One or more test leaders carry out the test procedure according to the requirements and the procedure decided upon. The data collected in the testing is found both as knowledge possessed by the test leader and as metrics and qualitative data in spreadsheets (see Figures 5.2 and 5.3 for more detailed examples). Figure 5.2 illustrates the "Raw" data from a series of tests. The different tabs in the spreadsheet contain all of 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. Figure 5.3 represents a Structured Data Summary, which was created and developed by Gary Denman, UIQ [106], where the qualitative findings of the testing are stored. They show 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 can be stored as comments in the spreadsheet.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	STUDIE	USER_ID	TES	UC_ID	TEE_1	TEE_2	TEE_3	TEE_4	TEE_5	TEE_6	TEE_1	TEE_2	TEE_3
2	BASE	F1:1	643	1.6	5	1	5	1	5	1	5	5	
3	BASE	F1:1	643	6.1	5	1	5	1	5	1	5	5	
4	BASE	F1:1	643	1.5	5	1	5	1	5	1	5	5	
5	BASE	F1:1	643	1.7	5	1	5	1	5	1	5	5	
6	BASE	F1:1	643	1.7	5	1	5	1	5	1	5	5	
7	BASE	F1:1	643	6.1	5	1	5	1	5	1	5	5	
8	BASE	F1:1	643	6.9	5	1	5	1	5	1	5	5	
9	BASE	F1:1	643	9.1	5	1	5	1	5	1	5	5	
10	BASE	F1:1	643	6.9	5	1	5	1	5	1	5	5	
11	BASE	F1:1	643	1.6	5	1	5	1	5	1	5	5	
12	BASE	F1:1	643	9.1	4	2	4	2	4	3	4	4	
13	BASE	F1:1	643	1.5	5	1	5	1	5	1	5	5	
14	BASE	F1:1	643	1.6	5	1	5	1	5	1	5	5	
15	BASE	F1:1	643	6.1	5	1	5	1	5	1	5	5	
16	BASE	F1:1	643	6.9	5	1	5	1	5	1	5	5	
17	BASE	F1:1	643	1.7	4	2	3	3	3	4	4	4	
18	BASE	F1:1	643	1.5	5	1	5	1	5	1	5	5	
19	BASE	F1:1	643	9.1	3	4	2	2	4	4	3	2	
20	BASE	F1:2	643	6.1	5	1	5	1	5	1	5	5	
21	BASE	F1:2	643	1.7	5	1	4	1	5	1	5	5	
22	BASE	F1:2	643	1.7	5	1	4	1	5	1	5	5	
23	BASE	F1:2	643	9.1	5	1	5	1	5	1	5	5	
24	BASE	F1:2	643	1.5	5	2	5	1	5	2	5	4	

Figure 5.2. Quantitative results in spreadsheets.



	A	B	C	D	E	F	G	H	I
1	User	1	2	3	4	5	6	7	8
2	UserID	M1:1	F1:1	F3:1	F2:1	F1:2	M2:1	F1:3	M1:2
3	X=XXXX, Y=YYYY, Z=ZZZZ Phone Order	K:N:W	N:W:K	K:W:N	W:K:N	N:K:W	W:N:K	K:N:W	N:W:
4									
5	XXXX								
6	UC1 - Receive and Answer an incoming call								
7	*Didn't know where to press*								
8	*Hesitated to press on screen*	y	y					y	y
9	*Tried to press HW under screen*					y			y
10	*Tried to press where says "Answer?"*								
11									
12	UC2 - Save the incoming call as a new contact - "Joanne"								
13	*Save Dialog*	y		y			y	y	y
14	*Name in call log doesn't update after saving number*				y				
15									
16	UC3 - Set an alarm for 8 o'clock tomorrow morning								
17	*Couldn't find alarms*	y	y	y	n		y	y	y
18	*Puzzled by time view - where alarms*								
19									
20	UC4 - Read an incoming SMS and reply with "I'm fine"								
21									
22	UC5 - Make a phone call to Joanne(07949 877249)								
23	*Looking for Green and Red*								
24									
25	UC6 - Create a new SMS - "Hi meet at 5" and send to Joanne								
26	*New SMS flow confusing - To: first*	y					y		
27	*How enter contact - when To: in focus - no "contacts"*	y							
28	*Thought space on zero*								
29	*Generally difficult to add contact*								
30									

Figure 5.3. Qualitative results in spreadsheets (product information removed).

The results of the testing are thereby a combination of the 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 backed up with the knowledge possessed by the test leader, who is the person who knows most

about the test process, the tester's reactions and explanations, and the context of the testing. Knowledge material is often presented verbally, but can if necessary be supported and confirmed by visual presentations of the data.

5.4 Agile and formal

The case study was grounded in thoughts concerning the importance of quality and agility in software processes, as specified previously in this chapter. We have always seen the importance of the framework as a tool for quality, and one purpose of the testing that this case study was based on was to verify this. Given the need for agility that has been identified, 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 evaluation framework. The following is the result of having studied the case study material from the perspective of the spectrum of different items that are taken up in the agile manifesto. See Section 2.9 for a detailed discussion of how the case study material was collected and analysed.

The agile movement is based on a number of core values, described in the agile manifesto [107], and explicated in the agile principles [108]. 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 [109] stresses that the intention is not to demolish the house of software development, which is 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 the items on the right are more likely to succeed in the long run. Even within the agile community there is disagreement about some of the choices, but it is accepted that discussions can lead to constructive criticism.

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 that different groups may have an interest in different factors of the evaluation 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 this emerging theory. Where the groups were found to fit on the scale is marked in bold text in 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.
- *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.
- *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 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.

- *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 being the concern of **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 would concern PO, so this is seen as being the concern of **D**
- *Following a plan* - From a short-term perspective, this is obviously 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.

5.5 On opposite sides of the spectrum

Our analysis of the case study material thus leads us to believe that "Designers", as in the agile manifesto, are interested in the items on the left hand side of the scale, rather than the items on the right (see Figure 5.4). 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"[24]. 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, certain actors 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 also still unsure exactly which roles in the development process belong to which group, but we are interested in looking at these extremes to see what their information requirements are in regard to the results of usability testing. Upon closer

inspection it may be found that none of the groups is actually on the far side of the spectrum for all of the points in the manifesto, and more work must be done to examine this distribution and division. The case study that is presented in Chapter 6 is a first attempt to make this distribution and these divisions clearer.

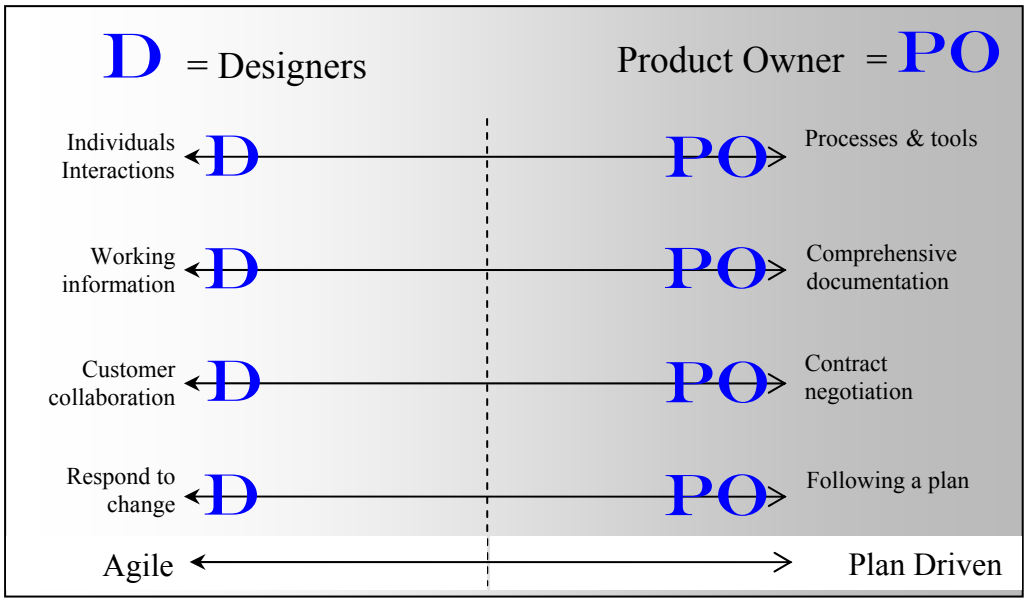


Figure 5.4. Groups and their diverging interests.

5.6 Discussion

In the following we discuss our results in relation to academic discourses in order 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 briefly upon two related academic discourses from the introductory chapter, i.e. the relation between quality and the need for cooperation between industry and research, and the relationship between quality and agility.

Since we are working 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 sufficiently flexible to accommodate the needs of many different stakeholder interests. The product must appeal to the broadest possible group, so it is problematic to have customers operating in dedicated mode with development team, with sufficient tacit knowledge to span the whole range of the application, which is what an agile approach actually requires to work best [110]. 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 [44]. Evidence suggests that drawing and learning from experience may be as important as taking a rational approach to testing [43]. 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 [110].

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 [110]. 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 [22] 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 [22] 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. [43]. 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 [43]. 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. This case study 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.

5.7 Conclusion and further work

In the evaluation framework, we have managed to implement a sufficient balance between agility and plan driven formalism that satisfies practitioners in many roles. The industrial reality that has driven the development of this

evaluation framework 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 evaluation framework demonstrates one way to balance demands for agile results with demands for formal results when performing usability testing for quality assurance. The evaluation framework 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 this chapter is a confirmation of 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 can be seen as both partly an illustration of the gap between industry and research, and partly an illustration of how agile approaches in practice 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 Mobile Communication in this case study gave 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 evaluation framework 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 managed to find a successful balance between agility and formalism that works in industry and that exhibits interesting qualities that can be of interest to both the agile and the software engineering community.

As a follow up to this case study, a further case study has been performed, collecting and analysing more information regarding the attitudes of Product Owners and Designers towards the type of information they require from testing and their preferred presentation formats, helping to define the groups and their needs, and helping us to place them on the map of the manifesto. Chapter 6 presents the results this from this related case study.

Future work based on this case study is to extend the study to other companies, to examine more closely the attitudes of different stakeholders, to clarify the division between Designers and Product Owners, and to gain more information on the aspects of agility and formality that are important in their everyday work. We already have a great deal of information regarding the shop floor perspective, and it will also be necessary to concentrate more closely on the perspectives of management, to discover how satisfy the formal needs of management, whilst still retaining the agility that already exists today and that has been found to be necessary.


chapter
six
Presentation
Methods
information
Designers
respondents
data
style
Product
Owners
results
test
two
method
Case
comparison
different
number
questions
presented
regarding
use
comments
whether
gives
ranked
also
found
PO
spreadsheet
diagram
early
time
needs
following
UIQ
contained
details
give
answer
see
questionnaire
factors
device
PowerPoint
project
experience
stakeholders
need
groups
replies
whilst
UTUM
three
system
issues
Curve
contains
analysis
whole
contextual
complete
identify
table
present
tests
question
user
seen
type
previous
detail
fact
one
verbal
answers
first
raw
taken
detailed
sheet
work
containing
NP
SDS
rankings
presence
given
Study
qualitative
findings
presenting
problems
usability
attitudes
Designer
best
help
respondent
leader
difficult
difference
process
however
total
development
design
testing
Preferred
Excel
possible
quantitative
Summary
roles
participants
stages
devices
appear
Figure
Factor
towards
points
rank
level
supported
company
brief
based
presentations
ways
find
means
asked
second
illustrates
made
differences
included
basis
group
way
important
although
used
needed
form
without
Excel

Chapter Six

6. Preferred presentation methods – a case study

This chapter concerns a case study that is performed as a follow up to the work in Chapter 5, regarding the balance between Agility and Formalism. We have performed a study to verify how our thoughts fit in with the actual situation as it is experienced by UIQ and their customers. It is an investigation of attitudes regarding which types of usability findings stakeholders need to see, and their preferred presentation methods. For a more detailed description of the methodology used in the case study, and a discussion of threats regarding reliability and validity, see Section 2.9.

As written in the previous chapter, the results of the case study appeared to show the existence of two general groups of stakeholders with different information needs, ranging from the group that we have called 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.

In an attempt 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.

The work in this study was performed 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 Chapter 5?
- 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?

In this chapter, we present results that indicate the needs of different actors in the mobile phone industry, that are a validation of the ways UTUM results are presented today. 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 who have different information requirements. These results point to future work; they give a basis for performing a wider and deeper study, and allow us to formulate a hypothesis for following up our results.

6.1 The questionnaire

A document was compiled that illustrated ten methods for presenting the results of UTUM tests. The choice of methods was made 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 were a selection of 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, and include a number of graphical representations of the data and results.

The document began with a brief introduction to the background to our study, and links to more information on the UTUM test for those who were unfamiliar with UTUM. We asked the participants for their help in our research in their roles as people who were representative in the design and development process. We gave a brief description of the task that we asked them to perform, and how the results should be returned to us.

On the following pages we gave an illustration of the ten chosen methods of presentation. We showed one method per page, and the illustration was complemented by a brief description of the presentation method and the information contained in it.

The participants were asked to read through the document and then complete the task by filling in their preferences in a spreadsheet. The spreadsheet contained three separate worksheets. The first contained step by step instructions on how to complete the task.

The second worksheet contained the table where the respondent should rank the methods. For each presentation method there was also a cell where the respondent could respond to the following statement: "*I would always need the test leader to explain this to me*". This question was included so that we could collect more information about the importance of the test leader in the testing and presentation procedure. The respondents were asked to prioritise according to a cumulative voting system, also called the $100 method. For more information regarding this, see Section 2.9. The respondent was asked to specify their role in the design and development process, by choosing one of a given number of roles. The roles specified in the list were also defined together with the test leader, and were:

- Product planning
- System Designer
- UI Designer
- System Verification
- Project Lead
- Sales and Marketing
- Management
- Other

If the respondent thought that their role was not included in the list, it was possible to choose "Other" and write the specific role in the cell below. In this worksheet, it was also possible for the respondent to write whether she or he thought that there was any form of presentation that was missing amongst the listed alternatives, and where she or he could describe the method that they would like to have seen.

In the third worksheet, respondents could write comments regarding the UTUM test or our questionnaire. In particular we asked for comments regarding changes in their favoured methods of presentation at different stages in the design and development process.

The mail, which contained the above documents as attachments, contained the following text:

> Hi
>
> My name is Jeff Winter, and I would like to ask you for some assistance. As a researcher at Blekinge Institute of Technology (as well as colleague at UIQ Technology) I would be grateful if you could help us in our research collaboration.
>
> I have been working together with UIQ Technology in the development of the UTUM test, and we need to find the answer to some questions regarding the presentation of test results. To help find these answers, I would appreciate it if you could take the time to answer a number of questions. I enclose two files: "research survey UTUM presentation.pdf", and "UTUM Prioritisation of presentation methods.xls". The first file includes information regarding the background to the task that we would like you to help us with, instructions for how to proceed, and the questions. The second file is where I would like you to write your answers to the questions that we have posed.
> We would be very grateful if you could take some time to read through the documents, answer the questions, and then return the answers to us, by mailing the spreadsheet to me at: jeff.winter@bth.se
>
> If you have any questions, do not hesitate to ask me.
>
> Please send your reply to me by 2nd June 2008 at the latest.
>
> Regards,
>
> Jeff Winter

The mail and attachments were sent to 29 people, mostly within UIQ but also some people from UIQ's licensees. The list of participants was 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.

6.2 Results of the questionnaire

In total, we received ten replies to our questionnaire. Few respondents, a total of six**,** had replied to the questionnaire within the stipulated time, so one day after

the preliminary deadline, we sent out a reminder to the participants 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. This means that we received ten 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.

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

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

As can be seen, there were no replies from the following categories:

- System Verification
- Project Lead
- Sales and Marketing
- Management

This of course means that it is impossible to give complete answers to the research questions that this study was intended to answer, although it should still help us to answer some of the questions, and give some indications that give us a better understanding of factors that affect the answers to the other questions. This work can be seen as helping us formulate hypotheses for further work regarding these questions.

We have attempted to divide the respondents according to the tentative schema found in the previous chapter, between Designers (D) and Product Owners (PO). Some respondents, in particular those who worked with system design, were difficult to place in a particular category. The actual 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. This was done to establish where the respondents were placed in the design and development processes. 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.

We found that five respondents could be assumed to belong to the Designer group. These were the two UI designers, the usability specialist, and two of the system designers. The remaining five respondents could be seen as representatives of the group of Product Owners. These were the two members

of product planning, the respondent from the CTO office and two of the system designers. In the following, where a specific respondent is referred to, the category they belong to is included in brackets after they are presented.

6.3 Presentation methods

Before we present any analysis of the data collected in the study, we give a brief description of the presentation methods that were included in the questionnaire.

Method 1: The Structured Data Summary (the SDS) [106]. A spreadsheet, containing 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 can be stored as comments in the spreadsheet.

Method 2: A spreadsheet containing all "raw" data. All of the "Raw" data from a series of tests. The worksheets in the spreadsheet contain the metrics 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 in the previous method. A very brief written explanation of the findings is given.

Method 6: Comparison of two factors (more in depth details). The same image as in the two previous methods. In this case there is a more extensive explanation of the results, and the findings made by the test leader. The test leader has also written some 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 but not presented by the test leader. In this method, 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, produced and presented 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 is the type of presentation that 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 collected data, 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.

6.4 Data analysis

The numerical data that were collected in this study are presented in an appendix (Appendix B, Table B.1). For the 10 respondents in the study, the number of methods ranked per individual ranges between 4 and 10. The mean points allocated per method ranges between 25 and 10. The mean number of choices made was 6.7 and the median was 5.5. The spread of points allocated was between 5 and 70 (respondent 5) and 9 and 16 (respondent 4).

Figure 6.1 is a box and whisker plot that shows this 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 the analysis that follows, we use various ways of summarising the data in the spreadsheets that were completed and returned. In order to gain a composite picture of the respondents' attitudes, the different 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.

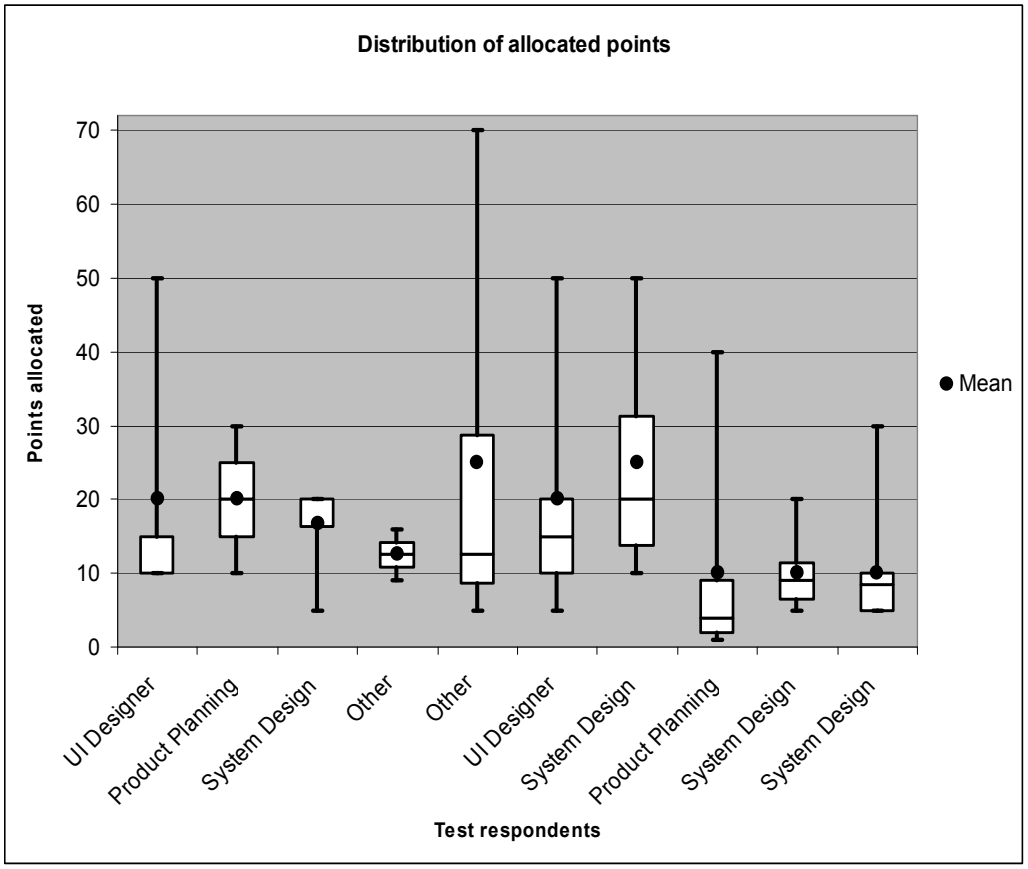


Figure 6.1. Distribution of points allocated per respondent

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 (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.

To gain a composite picture of which methods are most popular, we calculate an overall rank, by adding the positions awarded to the different methods, e.g. a method has ranks of 1, 1, and 3 in the different ranking schemes above, and therefore has a sum of 5. In this case, a lower sum is best, reflecting the fact that the method has been ranked highly in the different schemes.

Preferred methods: All respondents

Figure 6.2 illustrates the relationship between the methods and their ranks according to the calculation presented above.

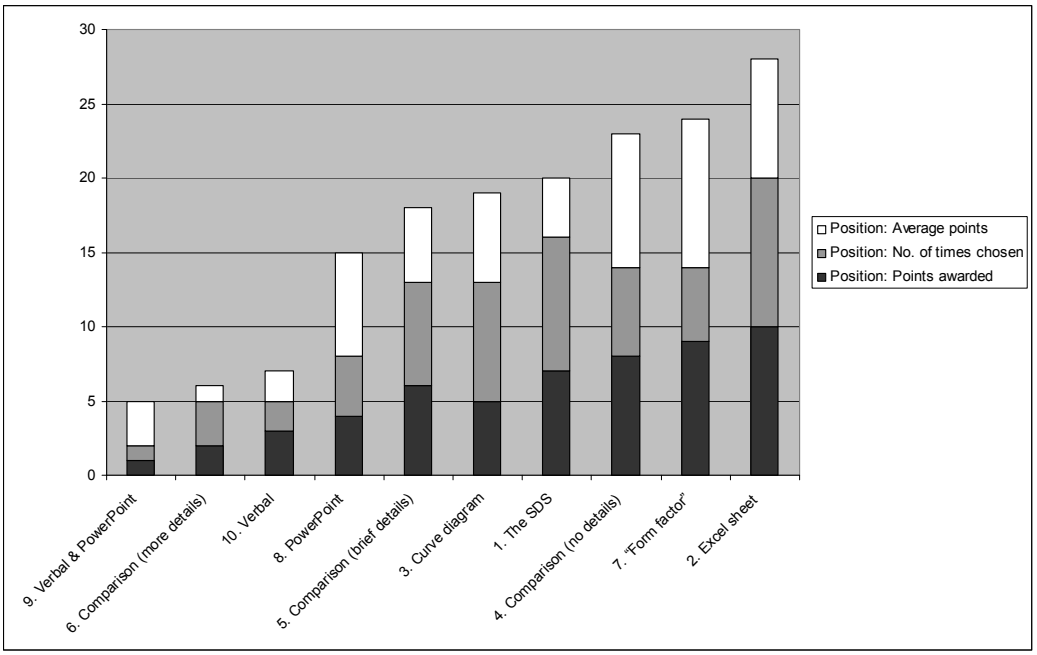


Figure 6.2. All respondents: Total rank (lowest point is best)

This gives the following:

1. Method 9: Verbal presentation supported by PowerPoint
2. Method 6: Comparison of two factors (more in depth details)
3. Method 10: Verbal presentation of early results
4. Method 8: PowerPoint presentation, no verbal presentation
5. Method 5: Comparison of two factors (brief details)
6. Method 3: A Curve diagram
7. Method 1: The Structured Data Summary (the SDS)
8. Method 4: Comparison of two factors (basic version)
9. Method 7: The "Form Factor" – an immediate response
10. Method 2: A spreadsheet containing all "raw" data

Total rank according to this calculation is very similar to ranking according to points allocated, and only two methods (ranked 5 and 6) have swapped places. Three methods appear to head the list. Two are verbal presentations, one being supported by PowerPoint and the other purely verbal. The fact that Method 2, a spreadsheet presenting "raw" data, is at the bottom of the list is interesting, as the method was actually chosen by two people as their second choice. One of these was a UI designer (D), and one was a System designer (PO). The others who ranked Method 2, although not before ninth place, were a system designer (D) and a product planner (PO). This suggests that this type of data is difficult to analyse and draw conclusions from, but that for those who are used to working with spreadsheets and who have the knowledge and possibility to make use of the data, there are many ways of utilising this data.

In order to identify the existence of differences between Designers and Product Owners, we continue with an analysis of the data on the basis of which group the respondent belongs to, to enable a comparison between the two groups. We are aware of the fact that the small number of respondents means that the comparison is based on limited mass of data, but believe that the data still gives an opportunity to draw some conclusions. We begin with the Designers.

Preferred methods: Designers

Figure 6.3 illustrates the relationship between the methods and their rankings. Here there appear to be two methods, scoring 4 and 5 points, which head the list: the Verbal presentation supported by PowerPoint and the Curve Diagram.

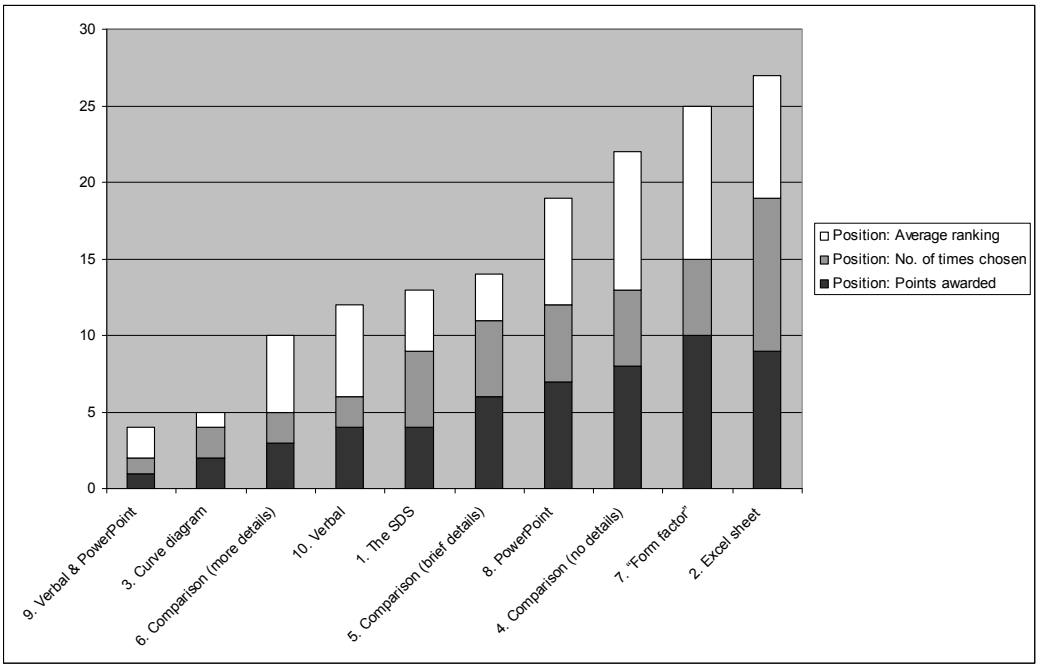


Figure 6.3. Designers: Total rank (lowest point is best)

Here too, results for total rank and position according to points awarded correlate closely. For Designers, the ranking of the methods of presentation:

1. Method 9: Verbal presentation supported by PowerPoint
2. Method 3: A Curve diagram
3. Method 6: Comparison of two factors (more in depth details)
4. Method 10: Verbal presentation of early results
5. Method 1: The Structured Data Summary (the SDS)
6. Method 5: Comparison of two factors (brief details)
7. Method 8: PowerPoint presentation, no verbal presentation
8. Method 4: Comparison of two factors (basic version)
9. Method 7: The "Form Factor" – an immediate response
10. Method 2: A spreadsheet containing all "raw" data

Before comparing these results with the group of respondents as a whole, we analyse the results for the group of Product Owners.

Preferred methods: Product Owners

Figure 6.4 illustrates, for Product Owners, the relationship between the methods and their ranks.

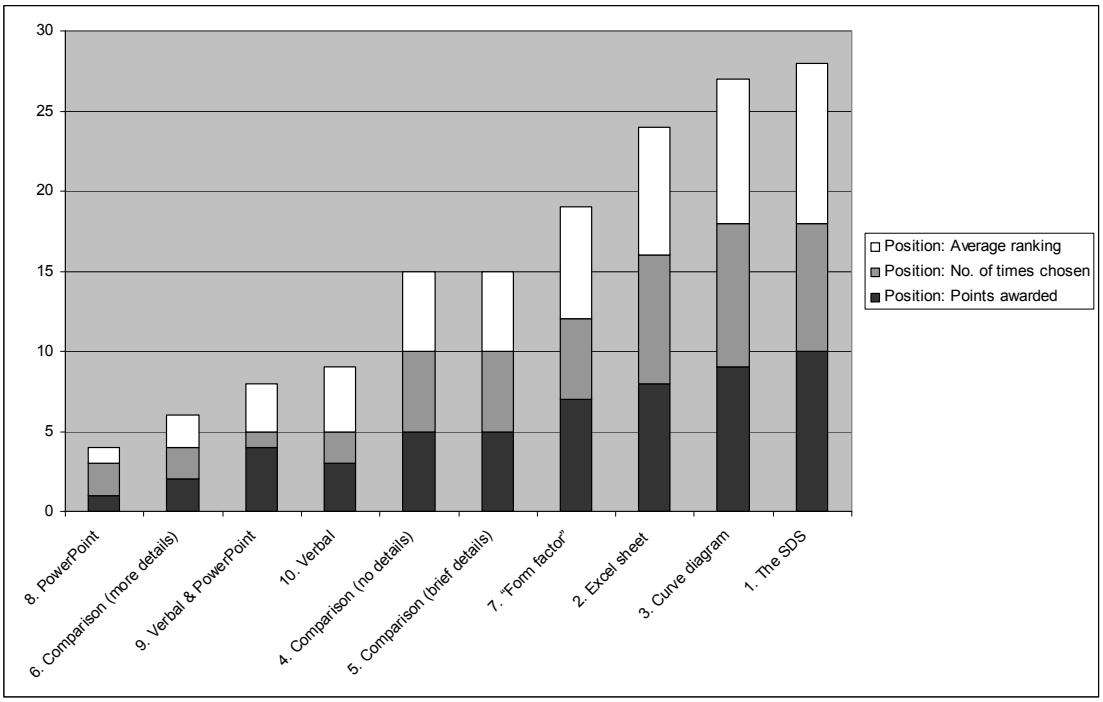


Figure 6.4. Product Owners: Total rank (lowest point is best)

Here, once again, there is little discrepancy between the total rank and the position according to points awarded. Thus, for Product Owners, the ranking of the methods of presentation is as follows:

1. Method 8: PowerPoint presentation, no verbal presentation
2. Method 6: Comparison of two factors (more in depth details)
3. Method 9: Verbal presentation supported by PowerPoint
4. Method 10: Verbal presentation of early results
5. Method 4: Comparison of two factors (basic version)
   Method 5: Comparison of two factors (brief details)
6. –
7. Method 7: The "Form Factor" – an immediate response
8. Method 2: A spreadsheet containing all "raw" data
9. Method 3: A Curve diagram
10. Method 1: The Structured Data Summary (the SDS)

Comparison of results for different groups

Having presented these results, we now turn to examining the differences between the results found in the different groups.

Figure 6.5 is an illustration of the differences in rankings between the two stakeholder groups. The first column shows, for each method, how it was ranked by the respondents as a whole. The second column shows how Designers ranked the method, and the third column shows how Product Owners ranked the method.

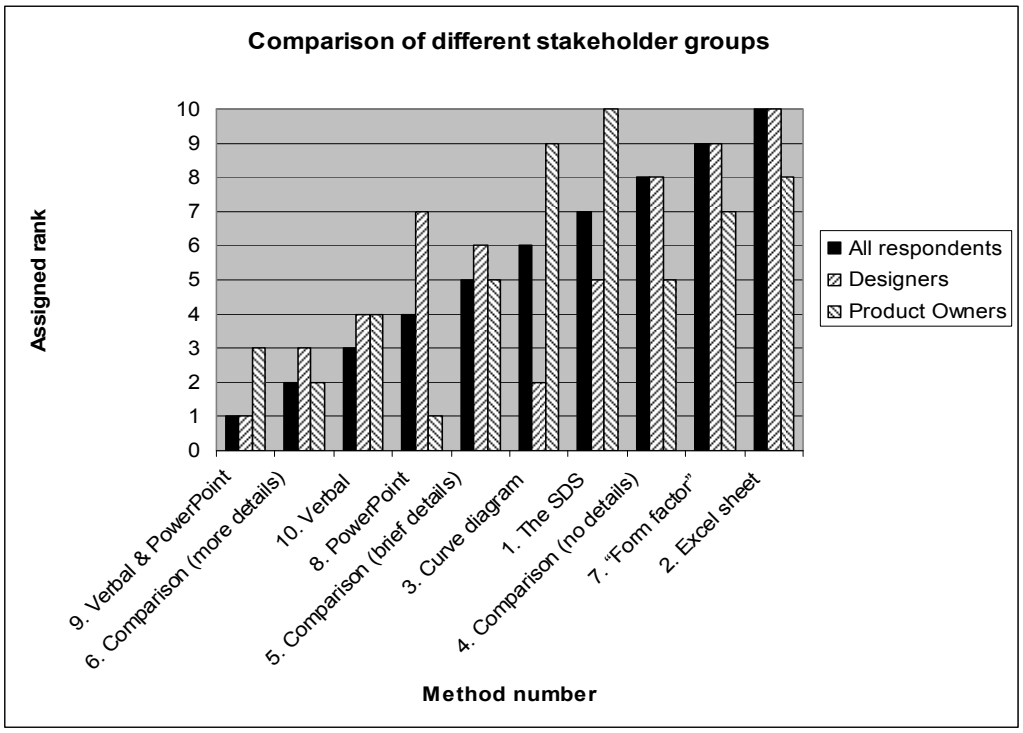


Figure 6.5. Comparison: All, Designers & Product Owners (lowest point is best)

Table 6.1 presents the methods in the order that they were ranked by all respondents. Cases where the opinions differ significantly between Designers and Product Owners (a difference of 3 places or more) are marked in bold italics, and will be the subject of discussion, to see whether we can draw any tentative conclusions about the presentation requirements of the different stakeholder groups.

Four methods, 1, 3, 4 and 8, are the subject of further analysis. We analyse them in the order of the greatest difference between the two groups.

There are no specific comments in the answers to the questionnaire regarding these particular methods, giving clues to why these differences occur. 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 following analysis will be based on the knowledge we have of the conditions at the company and the context where they worked. To verify these results, further studies would be needed.

Method	Rankings			Difference between groups
	All respondents	Designers	Product Owners	
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

Table 6.1. Comparison of ranks: All, Designers and Product Owners

Method 3: The Curve Diagram. Here there is a difference of 7 rankings. Designers ranked this as number 2, whilst Product Owners ranked it as number 9. The curve diagram is a way of illustrating the time taken to complete a use case, and is a comparison of several different devices. It gives an indication of which device is most efficient for performing the use case, and the shape of the curve shows when users finished the operation. There is information regarding the use case in question, the devices used, and the time taken, but there is no contextual information, explaining what the test leader observed during the testing.

The fact that Designers ranked this presentation so highly can be explained by the fact that if the diagram 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. The time taken to perform the use case can give an indication of the performance of the device, which can also be correlated with user satisfaction. The shape of the curve can also illustrate when problems arose during the use case. Thus, 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.

The reason Product Owners ranked this method so poorly is that the information is too detailed, on the level of an individual use case, whilst they as a group need to receive information about the product or device at a coarser level of detail; that is general and 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.

Method 8: **PowerPoint presentation, no verbal presentation**. Here there is a difference of 6 rankings, where Designers ranked this as number 7 and Product Owners ranked it as number 1. This is a presentation that is produced by the test leader, presenting the results graphically and in writing. This presentation needs some time to compose, as it is dependent on the analysis of the total mass of data collected in the study, but it is not intended to be presented by the test leader. This means that the level of detail in the presentation is complex.

This presentation can contain several ways of presenting the results of testing, and we believe that the lack of contextual information and the lack of opportunity to ask follow-up questions is the main reason that Designers would find this type of presentation of limited use. It gives a lot of information, and a broad view of the results of the testing, but does not contain sufficiently detailed information to allow Designers to identify the problems found or make decisions about solutions to problems. Without detailed information about context and what actually happened in the testing situation, it is difficult to interpret differences between devices, to know which problems there are in the device that is being developed, and it is thereby difficult to know what to do about the problems. The length of time taken to produce the presentation also means that it is less suitable for Designers, who are concerned with fixing product issues as early in the development process as possible. We 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.

We also believe that the reason that this type of presentation is of primary interest to Product Owners is that 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 an adequate amount of text, and gives an indication of the status of the product. It is also a presentation that is adapted to viewing without the presence of the test leader, meaning that 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)**. Here there is a difference of 5 rankings, where Designers ranked this as number 5 and Product Owners ranked it as number 10. The SDS contains the qualitative findings of the test, for each tester and each device, for every use case.

We believe that 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 every use case, there is information on issues that the test leader 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 under the title of the use case, and if there are many user comments registered in the spreadsheet this also gives an indication that there are issues that need to be investigated further. The contextual information contained gives clues to problems and issues that must be dealt with and gives hints for 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 for several reasons. Product Owners do not deal with products on the level of individual use cases, and this presentation gives detailed information on use cases, but is difficult to interpret for the device as a whole. The information is too finely detailed and is not adapted to the broad view of the product that the Product Owners need. The contextual information, which is spread throughout the spreadsheet, is also difficult to summarise and does not give a readily understandable of the device as a whole. This means that Product Owners find it difficult to make use of the information contained in this spreadsheet and thereby rank it as least useful for their needs.

Method 4: Comparison of two factors (basic version). Here there is a difference of three rankings, where Designers ranked this as eight whilst Product Owners ranked it as five. This method illustrates a summary of a series of tests, showing a comparison between satisfaction and efficiency.

Once again, we believe that the lack of detail and of contextual information means that the Designers find it difficult to read information from the diagram that would allow them to identify problems with the device. It simply provides them a broad 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. For Product Owners 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 the above, 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.

The fact that Designers ranked Method 3 so highly could be seen as rather surprising, given that the lack of contextual information does not allow them to gain an understanding of the reasons for differences between devices. However, the information contained in the diagram, if interpreted properly, actually gives a wealth of information on the use case that can be used to identify problems in the product.

Later in this chapter, we examine whether there are differences between the two groups concerning attitudes towards the presence of the test leader when presenting the results of testing.

Missing methods

To continue our analysis, we examine whether the data suggests that there are methods that the respondents feel are missing amongst the methods that we presented. Four respondents wrote comments about methods that they thought were missing and would like to see. One respondent, involved in product planning (PO), wanted to see methods more dedicated to measuring aspects concerned with user experience. This is not merely a case of needing new ways of presenting the information; it also concerns a shift regarding the type of testing that must be performed, and the types of data that must be collected. As previously noted, there is tendency within the industry to move away from usability towards an emphasis on User Experience, and this answer reflects this movement.

One of the System designers (D) stated that what was missing was a single method that combined all of the given methods. The usability expert (D) also expressed the wish to see a combination of the different methods. One of the respondents, from the CTO office (PO), saw no need for Method 7, the hardware evaluation. He saw these aspects of hardware design as being the responsibility of the producer and manufacturer of the device, whilst UIQ as a software company are primarily responsible for the design and development of the software.

Once again, given the limited number of replies, it is difficult to draw any conclusions, but this suggests that taking into account aspects of user experience has become more important in the design and development process, and that there is a need for some type of presentation method that combines the positive features of all of the above 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. Concerning the hardware evaluation, it is can be argued that this also gives useful information, for several reasons. The first is that it can give an indication of why some users experience dissatisfaction with a product. For one device, for example, it was found that none of the female testers were satisfied with the device as hardware, and this is naturally reflected in the judgement of the device as a whole. It is also useful information that can be passed on to partners in the development process, since the company has been cooperating with hardware producers.

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, a UI designer (D), stated that the answers were a "shopping list" of the information they would like to get from UTUM, and 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 (Method 10) and verbal presentation supported by PowerPoint (Method 9), whilst a non-verbal presentation, in conjunction with the "raw" data in the spreadsheet and the SDS would be more appropriate in 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, from system design (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 "raw" data and the SDS would become more appropriate, because the problem is understood and the quantitative answers are then more easily interpreted than the qualitative answers.

Another respondent, from product planning (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 may 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.

Attitudes towards the role of the test leader

To finish this part of the data analysis, we examine attitudes towards the role of the test leader in the presentation process. 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.

Since two of the presentation methods presuppose the presence of the test leader, some of the results were obvious; these two methods are therefore excluded from the analysis that follows. The first of these is Method 9, which is listed as being a verbal presentation supported by PowerPoint. Even here, one respondent (D) found the presence of the test leader as being unnecessary. This could be interpreted as an opinion that the presentation in itself gives enough information, and this respondent had not prioritised this type of presentation highly. The second is Method 10, which is a verbal presentation of early results.

Two of the respondents supplied no answers to this question, and one of the respondents only supplied answers regarding methods 9 and 10, so if we exclude these three respondents from the summary, there were a total of seven

respondents, of whom 4 gave answers for all eight methods, 1 gave five answers, and 2 gave three answers. The three respondents who did not answer these questions were all Product Owners, meaning that there were 5 designers and 2 Product Owners who answered these questions. See Table 6.2 for the results.

Method No.	*Test leader needed*	*Test leader not needed*
1	4	2
2	4	2
3	2	3
4	2	3
5	2	3
6	2	3
7	5	1
8	2	3
9	7	1
10	8	0

Table 6.2. Is the test leader needed to present the results?

In the analysis of each method below, the results are summarised to show how many answers were received, and how many Designers and Product Owners found the presence of the test leader necessary or unnecessary e.g.: (6 replies) (P=4: D=3, PO=1) (NP=2: D=2, PO=0), meaning that six respondents supplied answers, and that four respondents thought the test leader should be present (P=4), (three Designers and one Product Owner), whilst two respondents thought the test leader need not be present (NP=2), (two Designers and no Product Owners).

The first two methods are both spreadsheet based, but the spreadsheets contain different types of data.

Method 1: (6 replies) (P=4: D=3, PO=1) (NP=2: D=2, PO=0)

Method 2: (6 replies) (P=4: D=3, PO=1) (NP=2: D=2, PO=0) Two of the respondents, both Designers, changed their answers compared to Method 1, perhaps reflecting their personal preferences, given the fact that Method 2 contains quantitative data rather than the qualitative data in Method 1.

Method 3: (5 replies) (P=2: D=2, PO=0) (NP=3: D=3, PO=0)

Method 4: (5 replies) (P=2: D=0, PO=2) (NP=3: D=3, PO=0)

Method 5 (5 replies) (P=2: D=0, PO=2) (NP=3: D=3, PO=0)

Method 6: (5 replies) (P=2: D=1, PO=1) (NP=3: D=3, PO=0)

Method 7: (6 replies) (P=5: D=4, PO=1) (NP=1: D=0, PO=1)

Method 8 (5 replies) (P=2: D=1, PO=1) (NP=3 D=3, PO=0), even though this is specified as being a PowerPoint presentation without verbal presentation, there were still 2 respondents who thought that it was necessary for the test leader to participate.

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.

6.5 Discussion

There are several threats regarding reliability and validity, and for a more detailed discussion of these, see Section 2.9.

We now begin by returning to the five research questions posed in this case study:

- 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 Chapter 5?
- 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 answer to the first question, whether any presentation methods are generally preferred, is that verbal presentations are in general preferred by the respondents as a whole. 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, whether we can identify differences between the two groups of stakeholders, 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. 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 thought the test leader should be present whilst Designers did not.

The answer to the third question, whether there are methods that are lacking in the current presentation methods, was found taking into account and visualising aspects of user experience is becoming more important in the design and development process, and although the question was not posed in this fashion, this indicates that testing must be adapted to capture these aspects more implicitly. It is also apparent that there is a need for some type of composite presentation method that combines 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.

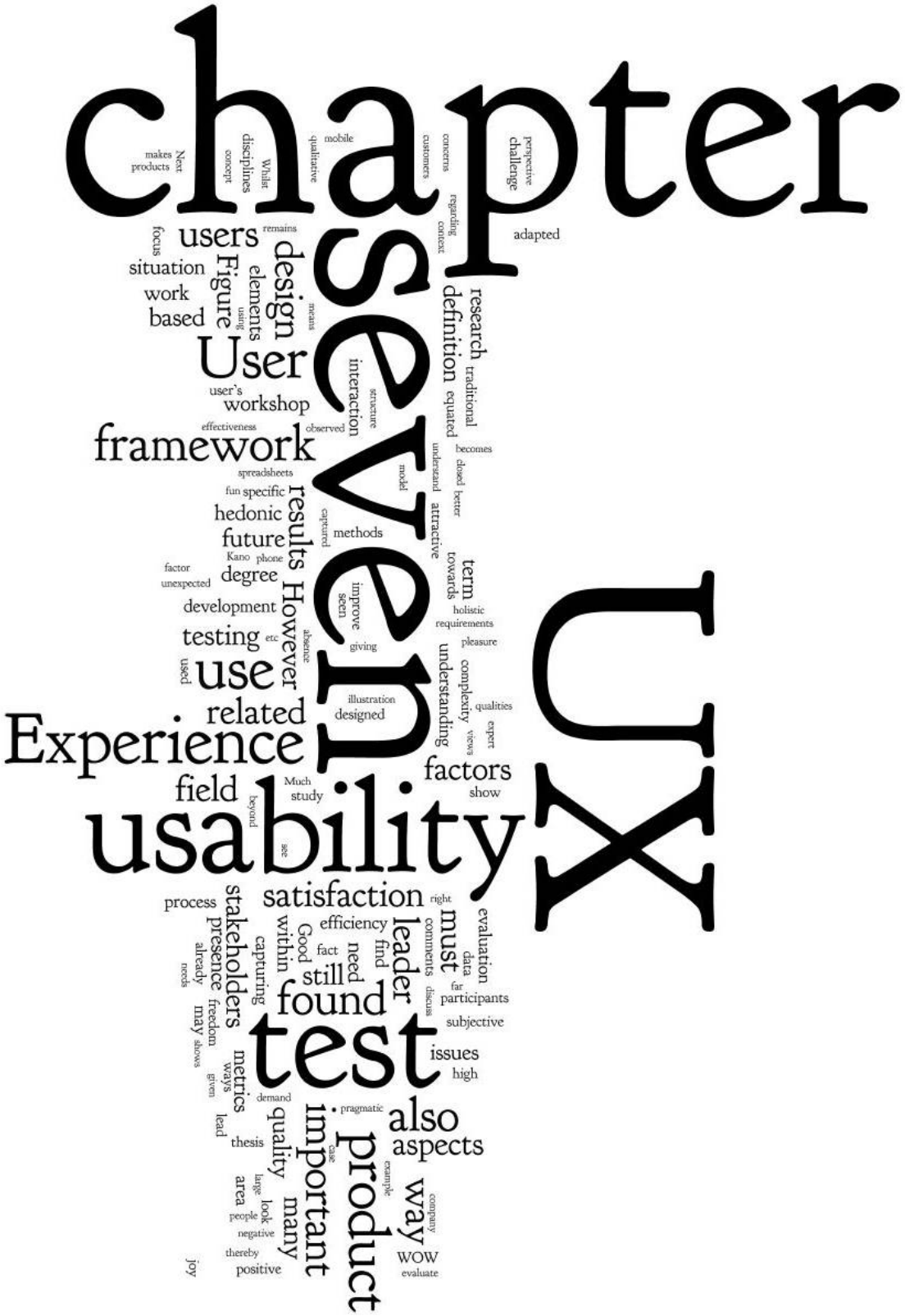
No clear answers can be found for the fourth question, whether information needs, and preferred methods change during different phases of a design and development project. The limited replies that were given suggest however that the required presentation methods do change during a project, with the more verbally oriented and qualitative presentations being important in early stages of a project, in the concrete practice of design and development, and the more quantitative orientated methods being important in later stages and as a 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 conclude, the work presented here is a preliminary study that indicates the needs of different actors in the telecom industry, and it is a validation of the ways in which UTUM results have been presented. It provides guidelines to improving the ways in which the results can be presented in the future. It is also a confirmation of the fact that there are different groups of stakeholders 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.

The next chapter in is a continuation of a theme that has arisen in previous chapters in this thesis, and a theme that has also appeared in this chapter. We now turn to the area of user experience and what we have called the Wow factor.


chapter
seven
UX
usability
Experience
framework
test
User
design
users
product
important
also
aspects
way
many
quality
found
leader
must
satisfaction
stakeholders
use
however
results
hedonic
future
factors
field
related
testing
development
degree
term
understanding
research
definition
interaction
workshop
Figure
elements
based
work
situation
focus
remains
adapted
perspective
challenge
concerns
customers
mobile
qualitative
Whilst
disciplines
concept
Next
makes
products
regarding
context
traditional
equated
becomes
closed
better
attractive
towards
holistic
requirements
pleasure
complexity
qualities
expert
views
show
illustration
designed
absence
etc
giving
improve
seen
methods
captured
Kano
phone
factor
unexpected
fun
specific
spreadsheets
model
understand
effectiveness
observed
structure
user's
means
using
Much
study
beyond
see
process
right
efficiency
evaluation
comments
data
find
need
fact
Good
within
capturing
presence
already
needs
still
far
discuss
participants
subjective
issues
high
may
shows
freedom
metrics
ways
given
demand
lead
thesis
pragmatic
case
example
large
area
people
look
negative
company
thereby
positive
joy
WOW
evaluate

Chapter Seven

7. UX and the Wow factor

This chapter is a discussion of how the evaluation framework, originally designed for measuring usability in a traditional sense, also provides a practical User Experience (UX) evaluation. Some of what follows is adapted from material published in the workshop paper: *Reporting User experience through Usability within the telecommunications industry* [111]. The chapter rounds off the thesis, and points out directions for future work, in the area of capturing (UX) and the challenge of communicating it adequately to actors in the organisation. The concept of the Wow factor arose during the cooperative research process, and is also connected to the concept of UX. This term is explained more fully later in the chapter.

Here, we follow up the introduction to UX that was given in Chapter 1, and discuss the connections between usability and UX, and how the evaluation framework is related to capturing not only usability but also to some degree UX. It also shows the position that UX held within UIQ at the time that the company closed, and discusses the need for development in the area that was apparent within the company at that time. It also provides a context, showing why the situation is as it is today.

7.1 What is this thing called User eXperience?

We begin by looking more closely at UX, how it is defined, and how it is related to the way that usability and quality have been defined and measured. In our work together with UIQ, the concept of UX arose more and more frequently, but when we introduced the term, no more detailed definition was given beyond the fact that UX is seen as a more encompassing term than usability.

Our uncertainty reflects the prevailing situation. Much work is being done within the field of UX, both theoretical and industrial, but much remains to be done, and it is still uncertain how UX differs from the traditional usability perspective [28], although if UX is to be accepted as a topic in its own right, it must differentiate itself from and add to the traditional view of interactive product quality [25].

Reflecting the fact that this is a new and developing field, there are a number of definitions of UX in existence. Still, there is a striking degree of consensus that UX is an area where there is no coherent and consistent definition of the field. Definitions of UX include the following:

- "*…the quality of experience a person has when interacting with a specific design. This can range from a specific artifact, such as a cup, toy or website, up to larger, integrated experiences such as a museum or an airpor*t" [112],

- "*…all aspects of the end-user's interaction with the company, its services, and its products. The first requirement for an exemplary user experience is to meet the exact needs of the customer, without fuss or bother. Next comes simplicity and elegance that produce products that are a joy to own, a joy to use. True user experience goes far beyond giving customers what they say they want, or providing checklist features*" [113].
- "*A consequence of a user's internal state (predispositions, expectations, needs, motivation, mood, etc.), the characteristics of the designed system (e.g. complexity, purpose, usability, functionality, etc.) and the context (or the environment) within which the interaction occurs (e.g. organisational/social setting, meaningfulness of the activity, voluntariness of use, etc)*" [114].

This lack of a coherent definition is a problem within the field. *COST Action 294 MAUSE* [115] is a research organisation under the umbrella of COST – European Cooperation in the field of Scientific and Technical research [116]. The goal of COST294 is to promote the use of scientific methods on Usability Evaluation Methods development, evaluation and comparison. The foreword to a report from the COST294-MAUSE workshop on User Experience in 2006, states that the field of UX is theoretically incoherent, and methodologically immature, and that there is no definition of UX or theory of experience that can inform the HCI community how to design for and evaluate UX [117].

Some of the same concerns were again noted in a workshop held at CHI'2008, where in the call for workshop papers it was stated that "*the definition of UX is not settled*", and that "*we need to find lightweight evaluation methods applicable for iterative prototype development"* [118]. The conclusions of this workshop also state that to evaluate UX requires an understanding of what UX actually is, and that this understanding is still far from settled [28].

The COST294 Newsletter from April 2008 [119] refers to a study that can still be regarded as work in progress [120], pointing to the fact that there is still a need for a standardized definition of UX. More than 100 researchers and practitioners have been invited to indicate to which extent they agree with the statement "There is a definite need for a standardized definition of the term UX". The results so far show that of 54 respondents, 59% agreed or strongly agreed with the statement, 19% were neutral, and 22% disagreed or strongly disagreed.

One illustration of the complexity of the area of UX is given by Dan Saffer [121], who shows the relationship between Interaction Design, Information architecture, Communication design, HCI and many other disciplines, and shows how most of the disciplines fall at least partially under the umbrella of UX Design (see Figure 7.1), characterised as the discipline of looking at all aspects of a user's encounter with a product or service, and ensuring that all of these are in harmony [121].

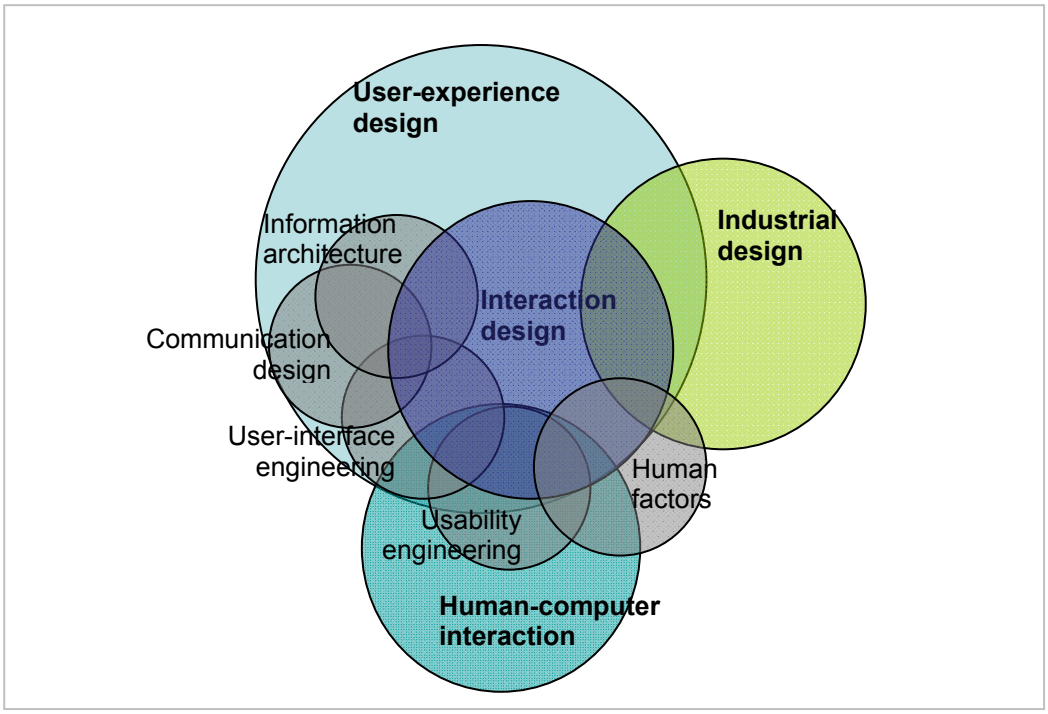


Figure 7.1. Overlapping disciplines of Interaction Design

Another way of showing the complexity of the field is found in Hassenzahl and Tractinsky, "User experience – a research agenda" [114], combining three perspectives: Beyond the instrumental; Emotion and affect, and; The experiential. Each of these contributes to an understanding of users' interactions with technology (see Figure 7.2). None of these single perspectives can however fully capture User Experience, and it is necessary to take all aspects into account.

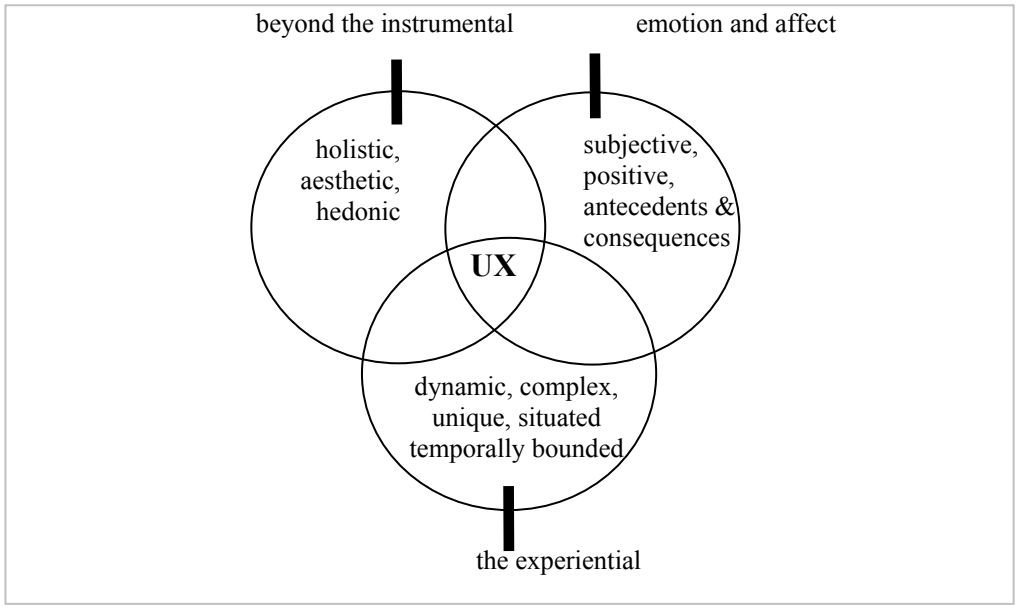


Figure 7.2. Facets of UX [114]

7.2 How are UX and usability related?

After this initial look at UX and the complexity that it entails, we return to the connections between UX and prevailing views regarding usability.

It is important to examine the similarities and differences between UX and usability. As already presented Chapter 1, usability has been concerned with pragmatic judgements, focusing on users and their tasks, based on observations of product use. Usability is equated with task related quality, based on efficiency, effectiveness and satisfaction [26]. Good product usability means that a user performs tasks in intended ways, in a way that does not lead to disturbance or annoyance. Usability testing is successful at identifying the factors whose absence may lead to dissatisfaction but whose presence does not necessarily lead to satisfaction. These factors can also be equated with the "must be" factors in the Kano model [15], which we discussed in Chapter 4 in this thesis. The field of usability has long experience of working in this fashion, and through this has come a long way towards satisfying industrial needs.

UX is seen as being holistic, subjective and positive [25]. An important term in UX is "hedonic", meaning related to, or characterised by, pleasure. UX takes a holistic approach, taking into account and finding a balance between both pragmatic aspects and hedonic aspects – non-task related aspects of product use and possession, such as beauty, challenge and joy. UX is thereby subjective, with an interest in the way people experience the products they use. Whilst both usability and UX can concentrate on efficiency, effectiveness and satisfaction, UX attempts to reduce satisfaction to elements such as fun, pride, pleasure and joy, and attempts to understand, define and quantify these elements [29].

It may be found that usability has been good at measuring the "must have" attributes that a customer expects, and even the qualities that are explicitly demanded by the customers, whereas UX is better at finding the "attractive attributes" that are unexpected and delight the customers by their presence.

7.3 So why is UX important?

The focus given to the negative that is found in the usability view of product quality is not regarded as unimportant in UX, but what is emphasised is that the positive is not simply the absence of the negative, and that the positive types of product experience connected with hedonic factors are important. Hedonic aspects are important, firstly because they colour the way that individuals experience owning and using a product, thereby affecting future behaviour, and secondly because they affect how people pass on to others their experience of owning a product. This is related to what sells well in the marketplace, and becomes more and more important in the type of market driven area where this study has taken place.

Returning once again to the Kano model, where the presence of attractive requirements can sometimes be found to outweigh a lack of "must have" features [15], we find that in much the same way, high levels of satisfaction result from the presence of hedonic qualities, rather than the absence of displeasers. When subjective factors such as beauty, fun, pleasure and pride, are

important, where a product can be seen as an expression of your personality or identity, capturing these factors becomes more important than being able to objectively measure usability. It becomes important to focus upon motivators and satisfiers. Technological advances in the field of mobile phones, enabling better graphics on larger screens, with better algorithms and processors, and longer battery life, have driven this movement towards hedonic qualities. Preconditions improve constantly, and users demand more and more as they get used to the improvements that are made.

As a result of this, whilst the focus in the future will not move away from the pragmatic, objective and negative views of product quality, and quality in use, there will also be an expansion towards holistic, subjective and positive views – from usability towards UX.

In the next section we discuss the connection between evaluation framework and UX.

7.4 Where do we find UX, and what do we do with it?

We have found that aspects of UX are currently captured in the evaluation framework. To a large degree this is through the presence of a usability expert in the testing situation. The test leader observes and judges usability and UX in specific use situations. Comments and impressions, both the test leader's and the participants', are stored in spreadsheets. This means that both usability and UX are already to be found in the complex mixture of metrics, qualitative data, and the mental picture in the mind of the test leader (see Figure 5.2 for an illustration of this). However, the results of the test are mostly translated to usability issues found, to optimize use case performance.

In the future, the handling of the results must however be adapted to emphasise UX. The question is how to utilise all this data and knowledge in the "big UX and Usability picture", in a situation where all of this knowledge is needed, and where many stakeholders want quantitative test results, whilst most of the information on UX is qualitative, and to a large degree is possessed by the test leader.

We have observed that some correlations between usability and UX that can be found in the metrics. Even though the metrics in the evaluation framework are designed to produce measures of usability, in some circumstances they can be analysed to show the presence of a good UX. For example, if there are high values for both satisfaction, as valuated by the test participant, and efficiency as evaluated by the test leader, then the high efficiency, rather than a good UX, may have increased the user's satisfaction. However, if the user evaluates satisfaction and effectiveness as high, whilst the test leader evaluates efficiency as low, this could mean that a good UX has raised the user evaluation. In this way, the metrics can be analysed and used to show findings for those stakeholders who demand metrics based results.

Comments recorded in spreadsheets also record what we have chosen to call the Wow factor, or delighters, and their opposites, the displeasers. The Wow factors can be equated with the Attractive requirements found in the Kano model, that

have the greatest impact on satisfaction regarding a certain product [15]. These Attractive requirements are not explicitly requested or expressed, but if present lead to an increased level of satisfaction. Comments recorded in the spreadsheets can concern both hardware and software issues and can be used to study which elements of the UX are pleasing or displeasing and thereby assist in improving UX.

Most of the UX factors that are found in the evaluation framework can be found to be related to measurements of specific use cases. It is however our experience that all kinds of subjects related to usability and UX, in the form of questions, comments and opinions, arise through the test leader's interaction with the participants. These concerns are important to the participants, and the test leader talks about them partly from a desire to understand why these issues are important, and partly to reveal users' suggestions for improving the current and future product.

This is related to the underlying philosophy of both the research and the evaluation framework, with its roots in participatory design, and using the techniques from ethnography. The test expert interacts and works together with the users to gain insight into how they experience being a mobile phone user, in order to gain an understanding of the user's perspective. The framework makes use of the inventiveness of phone users, giving them a chance to actively participate in the design process. The participatory design tradition respects the expertise and skills of the users, and this, combined with the inventiveness observed when users use their phones in real life situations, means that users provide important input for system development.

This way of working is necessary to really understand the behaviour of the users, to get to know why they behave as they do and what makes them do what they do. Through this, the test leader can act as a spokesman for users. The fact that the role of the test expert is directly integrated into the daily design process allows the continuous and direct use of user input in design and decision processes.

On the basis of the above, we cannot yet say that we, or the industry at large, have reached the level of successfully working with UX. The evaluation framework is an attempt to deal with a complex context, and to focus on visualising the results for stakeholders. In many ways it is successful, but there are still issues to be dealt with. The evaluation framework and the test leader capture and report usability flaws in the product. These are presented to the right stakeholders, and the issues go away. However we still find it difficult to address the product from a UX perspective, discovering what really makes people like or dislike a product. It is also difficult to know how to present this to the right stakeholders.

Thus the question remains of what we can do in the future to develop the evaluation framework and work more consciously with UX.

7.5 If UX is found in freedom, can we systematize it any further?

In the conclusions from the UXEM workshop, it was stated that there are already many methods in many disciplines that can be adapted to evaluate UX, and it is proposed that a road to understanding UX may be via a practice-driven development of the UX concept [28]. This is what we have tried to achieve with the evaluation framework. There are still many factors that would be interesting to study and discuss regarding the test and its connection to UX. We are interested in discovering ways that we can improve the use of the data already captured in the test. We also need to find things that we can do to improve the test. Can we for example leave the goal based nature of use case performance and introduce other elements, for example where the tester chooses use cases that are of particular interest to them personally, as in Swallow, Blythe and Wright's article on the UX of Smartphones? [122] In that case, participants were encouraged to reflect on the feelings and emotions that using a device evoked, by choosing five terms from a list of emotional adjectives, and then carrying out activities on their mobile phone that they felt were representative of such a description. Can we introduce other elements of experimentation or fun?

Figure 7.3 is an illustration of the structure of UX testing. Traditional usability testing demands an element of control, but testing UX may require a degree of freedom.

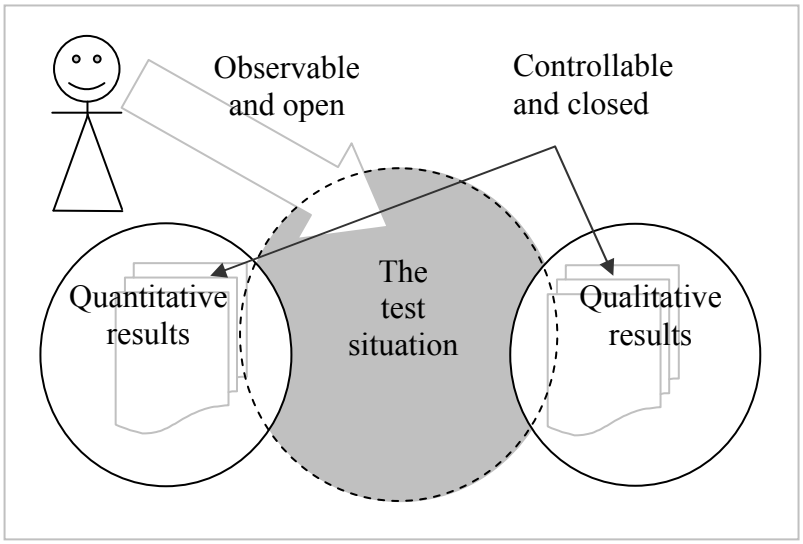


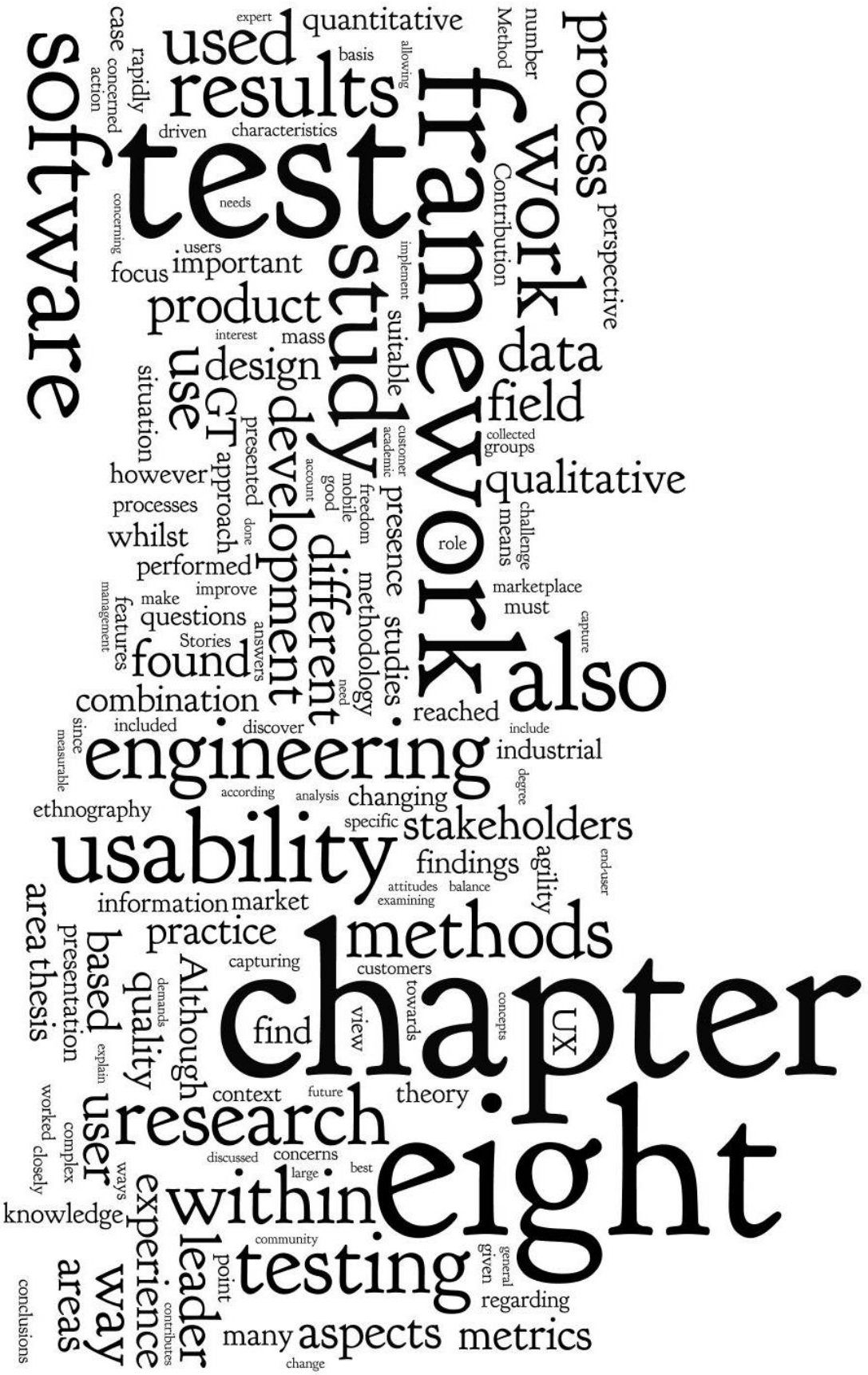
Figure 7.3. The structure of a usability test

Because of the many stakeholders in the process who demand easily presentable results based on metrics, there is a need for consistent measurable qualitative and quantitative data. This is represented by the closed circles in Figure 7.3. On the other hand, the test leader must be sensitive to the situation when and where the test takes place. The situation must be observable and open to the environment, symbolised by the dotted line around the circle, and also include elements from the closed circles. It should preferably be open for playing,

testing and experimenting. Testing UX must to a certain degree be equated with freedom from control, where the user, in the way he or she relates to the device being tested, expresses his or her personality and emotions. Here, the challenge concerns the job of the test leader, which must be capturing the surprising and the unexpected!

The User Experience is found mainly in judgements, in the “soft” side of the test, which captures feelings and unexpected aspects and this must be related to stakeholders in a process of direct or indirect communication. The challenge therefore remains of how we can systematize the results that arise from giving freedom a greater role in the testing procedure. We must also find ways of communicating the results that are captured in an acceptable way to the many different stakeholders in the organisations involved.

This brings us to the end of the thesis, and the next chapter contains a summary of the material that it contains, looking at the methodology used, the structure and use of the UTUM evaluation framework, the contribution that we have made to the field of software engineering and other fields, and a suggestion for future work based on the research that we have performed and the issues that we have observed.


chapter
eight
test
framework
software
study
work
process
used
results
quantitative
product
data
field
use
design
development
different
qualitative
presence
also
found
engineering
usability
methods
research
testing
within
experience
leader
way
areas
aspects
metrics
many
knowledge
user
based
practice
quality
Although
find
theory
context
future
concerns
discussed
capturing
customers
towards
concepts
UX
view
information
market
area
thesis
presentation
explain
worked
closely
complex
ways
large
best
community
point
regarding
given
general
contributes
conclusions
change
stakeholders
findings
agility
changing
industrial
ethnography
combination
reached
include
discover
included
since
measurable
according
analysis
degree
specific
attitudes
balance
examining
end-user
Stories
questions
features
management
make
improve
performed
whilst
processes
however
approach
presented
done
account
good
mobile
freedom
methodology
studies
role
means
challenge
marketplace
must
capture
answers
need
collected
groups
customer
academic
implement
suitable
interest
mass
situation
GT
focus
important
users
needs
concerning
driven
characteristics
rapidly
case
concerned
action
basis
allowing
expert
Method
number
Contribution
perspective

Chapter 8

8. Summary and conclusions

The principal focus of the work that is presented in this thesis is a usability evaluation framework for mass market mobile devices, allowing measurement, comparison and presentation of the usability and user experience of hand held devices. With its basis in the evaluation framework, and its use of quantitative measurements and qualitative judgments framed by a focus on usability testing, this thesis contributes to aspects related to capturing adequate real world usage in a continuously changing marketplace, and supporting different information needs in the cooperative process of building the software. The results are of practical and theoretical interest within the field of mobile phone development.

We begin this final chapter by summarising the concrete artefact that is the most obvious result of the research that this thesis is built upon: the evaluation framework itself, which was found to have a number of characteristics.

8.1 The framework

The framework was found to be quick and efficient. The testing phase was short, and the most important findings could be reported to the design process and included in the development process soon after the testing began. This was based on the conclusions reached by the test leaders. The continued recording and analysis process gave a statistical confirmation of the early findings, which was useful to supply stakeholders with metrics supporting the findings that had been reached.

The framework was able to handle complexity, and worked in a complex situation, with complicated relations between customers, partners, end-users and technologies. It was customer driven, and the features to be tested could be decided on the basis of customer requirements, or on end-user patterns. The test gives an end-user perspective to customers' questions regarding the usability and quality of their product, both from a software and hardware point of view.

The emphasis placed on metrics means that the test results can more easily find acceptance in the software development process. The amount of measurement, combined with the qualitative aspects of the test, gives a sound basis for verifying test results.

The above characteristics of the test, with its basis in qualitative research, usability principles, and metrics, show us how the framework can be used as a quality assurance tool in a wide perspective.

To sum up, the evaluation framework measures usability and elements of the user experience, and so measures aspects of quality in use. It has been tested in a complex industrial development project and has been found to be a valuable and flexible tool that is easy for a usability expert to learn and use.

We now continue by examining how our work contributes to the field of software engineering.

8.2 Our contribution to the field of software engineering

There are several features and aspects that are new and challenging in the evaluation framework that has been the subject of this study. These are: the mix of qualitative and quantitative methods in the framework, allowing it to target many different stakeholders; experience in applying these methods in the technology-focused and rapidly changing area of mobile phones; the challenge of addressing end users in a mass market; the challenge of finding presentation models for many stakeholders; making sure that the usability evaluation framework can be used in different stages in the industrial software development process.

To begin, we relate the research we have performed to three areas of software engineering: Market Driven Requirements Engineering (MDRE); Statistical Usage Testing (SUT); and Organisation and Product Together (OPT).

The framework has many features that contribute to solving problems that are discussed within the area of MDRE. It lets us formulate aspects of the marketplace that are hard to capture but are vital in a rapidly changing field, dependent on adapting to customers expectations and demands. The framework is in accordance with the goals of research within non-functional requirements, concerned with giving a common ground for grasping quality and communicating about it in different contexts.

SUT is concerned with finding usage patterns that result in failures whilst our testing is concerned with attitudes towards the behaviour and appearance of the device as a whole. Our testing reaches important results even if failures do not occur, since we observe the user who performs the use case. Whilst SUT can sometimes rely on suppositions about the views of users, our tests are based on actual use and the attitudes that accompany it.

Both our framework and OPT are based on an ambition to include the human side of the development. Our framework is however oriented towards both product and process, rather than process alone. It is also a simple process, whereas OPT is a more complex methodology.

The combination of factors listed above means that the evaluation framework solves a number of the problems that are found in these different areas within software engineering, especially in a rapidly changing marketplace, and it thereby makes a contribution that can improve practice in the field of software engineering, and that contributes to the theoretical work that is being performed within these different research areas.

There is a regrettable gap between software engineering and human-computer interaction (HCI). Although there is an impressive amount of user-centred methods within the HCI community, these methods are underused by software developers and organizations who find them difficult to implement. This may be because the methods are developed independently from the software engineering community. Empirical studies of software engineering have however demonstrated that successful usability practices do exist in software engineering organizations although they are not published, either at scientific

conferences or as organizational documents. There is thus a lack of publications and academic discourses on this subject within the field of software engineering.

Individuals with usability knowledge might find our implementations to be fairly standard 'usability'. This is true in the sense that we build heavily upon the research results of others within that field. However, there are features of our approach, in both the research methodology that we have used and the practice that we have developed, that differentiate it from the prevalent approaches, and that mean we make a contribution to the field. We discuss these features briefly below. The evaluation framework's specific combination of application area, theories and methods should be of interest to both industry and academia.

We continue by examining the methodology that we have used during the study.

8.3 The methodology

The research performed in this study has been a close cooperation between an academic and an industrial partner. It has been based on an action research approach, following the processes of Cooperative Method Development (CMD). A central aspect of action research is collaboration between researchers and those who are the focus of the research, and their participation in the process. CMD is a combination of qualitative social science fieldwork, with problem-oriented method, technique and process improvement. It consists of the three phases: Understanding Practice; Deliberate Improvements and; Implement and Observe Improvements.

This research has been based on a case study approach, which is a strategy where the focus is on a particular case, taking the context into account, and involving multiple methods of data collection. Although much of the data we collected in our case studies was qualitative, we also collected quantitative data, in much the same way as the work that we studied also consisted of the collection of qualitative and quantitative data. The techniques used included observation, interviews, and studies of documentation and project models.

Since we work in a software engineering context, the method of presenting the results differs from a traditional account of a case study, or ethnography. Here, the object of the report is not to allow the reader to experience being there, but is to make the results accessible to readers within the field of software engineering, and the risk of bias is therefore dealt with via a process of triangulation, member checking and the presence of an audit trail.

The study material has been analysed according to a grounded theory (GT) approach. GT is both a strategy for performing research and a style of analysing the data that arises from the research. GT is often interview-based, although other methods such as observation and document analysis can also be used, and it is applicable to a large variety of phenomena. Although we have not attempted to work according to pure GT practice, our study has relied to a large degree on the same basic methods of collecting data, although observation has perhaps had a more central role than interviews.

GT not only generates theory for the situation forming the focus for the study, but also grounds the theory in the data, where the interpretation is based on systematic enquiry. It is particularly useful in areas where theory and concepts that can explain what is going on are lacking. The rapid pace of change in the industrial context where we have worked means that theory and concepts concerning the field are also in a state of change, and GT was therefore a suitable methodology for us to use.

The work leading to this thesis has been inspired by ethnography. We have worked with ethnographic methods, and have been concerned with understanding the events in the everyday practice of the practitioners in the study. However, we have also, in accordance with the action research agenda, gone further than the descriptive side that is typical for ethnography. We have attempted to change practice, not only observe and understand it, and this has affected both our working methods and style of reporting the results. The departure from a descriptive approach is also reflected in the way in which we have used GT, which allowed a systematic analysis of the data in a way that is related to the prevalent way of working within SE. Here, GT was used to discover concepts and generate hypotheses from the fieldwork, which were then used to generate theory.

We have also worked with a participatory design (PD) perspective. Our use of PD has given participants the chance to make their voices heard. The principle behind this study has been to use the skills and knowledge of those who are most closely involved in the work, to design and implement a solution that is well designed, suitable for its purpose, and suitable in the organisation where it is to be used. These skills and knowledge have been both in the operational area, and in the area of organisational knowledge.

Thus, the range of tools and techniques used in this study has been extensive. Our main intention has been to capture and include the "members' point of view". The focus has been mainly on those who work "on the shop floor", although it was also important to include participants from management levels. All methods that have been seen as suitable for capturing and including the members' point of view have been seen as being useful. The combination of methods that have been in use in the research, and its combination of ethnography, and PD, is also reflected in the design of the evaluation framework itself. This particular mix of methods and techniques has been found to be a successful combination that has contributed to the development of a successful usability evaluation framework.

The study has been performed as an empirical study within industry. There are relatively few such studies in the field of software engineering, and it is important for the area of software engineering to increase the number of such studies, and to carry out more action research and case studies, to improve the industrial relevance of software engineering studies

The study presented here has spanned a number of areas, and in the following, we look briefly at some of the findings that have been reached within these areas and that have previously been presented in this thesis.

8.4 Findings in brief

The first area where our work is relevant concerns quality of use, and its focus on measurable aspects of a product, and the tasks and the context in which it is used, rather than on the product in isolation. This approach must be extended to cover the quality aspects related to the users' experience. The evaluation framework attempts to do this, whilst retaining measurable aspects of usability. It does this by collecting usability metrics and also metrics and knowledge of the user experience. Through the use of metrics, and judgements made by a test leader/usability expert, it begins to capture and measure the user experience, demonstrating quality from the customer and end-user point of view.

The evaluation framework provides another perspective, and becomes a way of verifying results reached via other methods. It emphasises quantitative aspects of testing, thereby enabling comparisons. It is flexible and easy to use when developing new designs, and when evaluating implemented designs to find possibilities for improvement. It finds product strengths and weaknesses in order to direct development resources to the relevant areas. It can provide an understanding of the usability of the product at a given time.

The evaluation framework is suitable for the general user of a mass market product, and also allows for a segmentation of users, and user groups, and allows the testing of specific functions on specific groups. It is a general purpose method that can be used for a mass market product, but can also be adapted for use for specific purposes.

The evaluation framework balances agility and plan driven formalism in a way that satisfies practitioners in many roles. Both formality and agility are vital when working in a rapidly changing mass market. Formal data is needed to support quick and agile results. The evaluation framework demonstrates how it is possible to balance these demands for both formal and agile results. It also supports an emphasis of "good organizational reasons" for not conforming to "best practice", since this type of testing is not according to "best practice" but is a complement that can meet organisational needs for a mass-market product in a rapidly changing marketplace, with many customers and end-users.

When studying how different stakeholders wished to see the results of usability testing, found that most stakeholders prefer verbal presentations. Concerning differences between the two groups of stakeholders, that we call Designers and Product Owners, 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.

Regarding the question of 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. However, spreadsheets containing test data require the presence of a test leader to explain the contents.

We have seen how taking into account and visualising user experience (UX) is becoming more important. Testing is moving towards a concern with UX, and UX is likely to affect the design and practice of the evaluation framework, and the presentation of results. Traditional usability testing demands an element of control, whilst testing UX may require a degree of freedom. Whilst there is a need for measurable qualitative and quantitative data, the test leader must also be sensitive to the situation when and where the test takes place. The situation must be observable and should preferably be open for playing, testing and experimenting. Testing UX must involve freedom from control, where the user expresses his or her personality and emotions. However, metrics programs are still a vital part of attempts to improve software processes. Thus, the framework is a tool for measuring usability, but given the presence of a usability expert, is also a step towards capturing and measuring user experience.

Our work includes a discussion of the need for agility, which has not so far been focused upon and discussed in connection with usability in the field of software engineering. We also contribute to the field, through the qualitative element of the evaluation framework, which is inspired by ethnography and participatory design.

The combination of all of the above, regarding the contribution the evaluation framework makes to areas within software engineering, the methodology used in the study, the construction of the evaluation framework itself, and the findings that we have reached in our study, means that the work performed in this study is a contribution of practical and theoretical interest within the software engineering community and the mobile phone branch.

We conclude this summary with suggestions for future work grounded in the findings from our study.

8.5 Future work

As a follow up to the summary and conclusions presented above, there are several areas where work remains to be done in the future.

One of these areas is the role of the test leader. The test leader is a key figure, with the knowledge to interpret the test results and their meaning, allowing her or him to present the results in appropriate ways. More research is required to discover what the characteristics of a good test leader are, how the test leader can support the processes within the companies, and in which way the test leader acts as an advocate of the user perspective, in line with the emphasis we have placed on participatory design and the members' point of view.

Work also remains to be done examining how the metrics from usability testing can be adapted to and used by different stakeholders in the design and development processes. It is important to find ways in which the metrics can be tailored to act as a boundary object between different disciplines. This is also connected to the role of the test leader, and how the test leader can interpret and present the metrics in the way that is most suitable for the situation and context.

Concerning the balance between agility and formality, the study should be extended to other companies, both large and small, to examine more closely the attitudes of different stakeholders, to clarify the division between Designers and Product Owners, and to gain more information on the aspects of agility and formality that are important in their everyday work. We have previously concentrated on the shop floor perspective, and must now concentrate more closely on the perspectives of management, to discover how to satisfy the needs of management, whilst still retaining the agility that already exists today and that has been found to be necessary. In line with the characteristics of the work previously performed, we feel that the continuation of 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.

UX is perhaps the most complex and immature area included in this study. There are still many factors that must be studied and discussed regarding usability testing in general and our evaluation framework in particular, and its connection to UX. We are interested in discovering ways that can improve the use of the data already captured in the test, to show the presence or absence of good UX. We also need to discover how to adapt the test to UX.

It is a challenge to systematize results that arise when freedom is given a greater role in the testing procedure, and to find ways of communicating the results that are captured in an acceptable way to the different stakeholders in the organisations involved. Here, the challenge concerns the test leader and the job of capturing the surprising and the unexpected.

8.6 Finally

To round off this thesis, we would like to quote the winner of the Nobel Prize in literature, 2008, Jean-Marie Gustave Le Clézio. In an interview for Swedish television, he was asked how he felt about giving answers to questions. He replied:

> "*Stories are good answers to questions. I have always thought that the best way to answer questions is to tell stories, since there are no proper answers to questions. Answers are found in life itself, or in stories, because they make the essence of life visible, in all its abundance, and its sustaining strength. So, I think it's better to ask authors to tell stories than answer questions.*" (Author's translation)

This thesis, written as it is as a monograph, and based as it is on countless hours of meetings, conversations, observations and a myriad of other sources, is our way of attempting to tell a story within the framework of the academic traditions. We would like to thank all of the people who have made this thesis possible, and hope that it tells at least a part of their story, in a way that they recognise.

References

9. Table of references

1. Rönkkö, K., *Interpretation, interaction and reality construction in software engineering: An explanatory model.* Information and Software Technology, 2007. 49(6): p. 682-693.
2. International Organization for Standardization, *ISO 9126-1 Software engineering - Product quality - Part 1: Quality model*. 2001. p. 25.
3. Grudin, J. and Pruitt, J., *Personas, Participatory Design and Product Development: An Infrastructure for Engagement*. in *Participatory Design Conference, 2002*. 2002. Toronto, Canada.
4. Rönkkö, K., Kilander, B., Hellman, M., and Dittrich, Y., *Personas is Not Applicable: Local Remedies Interpreted in a Wider Context*. in *Participatory Design Conference PDC '04*. 2004. Toronto, Canada.
5. Grudin, J., *The West Wing: Fiction can Serve Politics.* Scandinavian Journal of Information Systems, 2003. 15: p. 73-77.
6. Rettig, M., *Prototyping for tiny fingers.* Communications of the ACM, 1994. 37(4): p. 21 - 27.
7. Karlsson, L., Dahlstedt, Å.G., Regnell, B., Natt och Dag, J., and Persson, A., *Requirements engineering challenges in market-driven software development - An interview study with practitioners.* Information and Software Technology, 2007. 49(6): p. 588.
8. Regnell, B., Höst, M., and Berntsson Svensson, R., *A Quality Performance Model for Cost-Benefit Analysis of Non-functional Requirements Applied to the Mobile Handset Domain*, in *Requirements Engineering: Foundation for Software Quality*. 2007. p. 277.
9. Regnell, B., Höst, M., Natt och Dag, J., Beremark, P., and Hjelm, T., *An Industrial Case Study on Distributed Prioritisation in Market-Driven Requirements Engineering for Packaged Software.* Requirements Engineering, 2001. **6**(1): p. 51-62.
10. Pruitt, J. and Grudin, J., *Personas: Practice and Theory*. in *DUX 2003*. 2003. San Francisco, California.
11. Potts, C., *Invented requirements and imagined customers: requirements engineering for off-the-shelf software*. in *Requirements Engineering, 1995., Proceedings of the Second IEEE International Symposium on*. 1995.
12. Carlshamre, P., *Release Planning in Market-Driven Software Product Development: Provoking an Understanding.* Requirements Engineering, 2002. **7**(3): p. 139-151.
13. Jacobs, S., *Introducing measurable quality requirements: a case study*. in *Requirements Engineering, 1999. Proceedings. IEEE International Symposium on*. 1999.

14. Paech, B. and Kerkow, D., *Non-functional Requirements Engineering - Quality is Essential*. in *10th International Workshop on Requirements Engineering - Foundation for Software Quality - REFSQ 04*. 2004: Essener Informatik Beiträge Bd 9, Essen.
15. Sauerwein, E., Bailom, F., Matzler, K., and Hinterhuber, H. H., *The Kano Model: How To Delight Your Customers*. in *IX International Working Seminar on Production Economics,*. 1996. Innsbruck/Igls/Austria.
16. Sommerville, I., *Software Engineering*. 8 ed. 2007: Addison Wesley. 840.
17. Pfleeger, S.L. and Atlee, J.M., *Software Engineering*. 3rd ed. 2006, Upper Saddle River, NJ: Prentice Hall.
18. Regnell, B., Runeson, P., and Wohlin, C., *Towards integration of use case modelling and usage-based testing.* Journal of Systems and Software, 2000. 50(2): p. 117.
19. Cobb, R.H. and Mills, H.D., *Engineering software under statistical quality control.* Software, IEEE, 1990. 7(6): p. 45.
20. Seaman, C.B., *OPT: organization and process together*. in *Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: software engineering - Volume 1*. 1993. Toronto, Ontario, Canada: IBM Press.
21. BESQ. *Blekinge - Engineering Software Qualities*. 2008 [cited 2008-11-25]; Available from: www.bth.se/besq.
22. Osterweil, L., *Strategic directions in software quality.* ACM Computing Surveys (CSUR) 1996. 28(4): p. 738 - 750.
23. Harrold, M. J., *Testing: A Roadmap*. in *Proceedings of the Conference on The Future of Software Engineering*. 2000. Limmerick, Ireland: ACM Press.
24. Boehm, B.W., *Keynote address, 5th Workshop on Software Quality*. 2007: Minneapolis, MN.
25. Hassenzahl, M., Lai-Chong Law, E. and Hvannberg, E.T., *User Experience - Towards a unified view*. in *UX WS NordiCHI'06*. 2006. Oslo, Norway: cost294.org.
26. International Organization for Standardization, *ISO 9241-11 (1998): Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) - Part 11: Guidance on Usability*. 1998.
27. Tietjen, M.A. and Myers, R.M., *Motivation and job satisfaction.* Management Decision, 1998. 36(4): p. 226-231.
28. 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.
29. Lai-Chong Law, E., Hvannberg, E.T. and Hassenzahl, M.. *Foreword to User Experience - towards a unified view*. in *UX WS NordiCHI'06*. 2006. Oslo: cost294.org.

30. Bass, L. *Bridging the Gap Between Usability and Software Engineering (invited session)*. in *Proceedings of HCI International*. 2003. Crete, Greece.

31. Gulliksen, J. *Integrating User Centred Systems Design in the Software Engineering process (invited session)*. in *Proceedings of HCI International*. 2003. Crete, Greece.

32. Harning, M.B. and Vanderdonckt, J., *Closing the Gaps:Software Engineering and Human Computer Interaction*. in *Workshop at the 9th IFIP TC13 International conference on Human-computer Interaction (INTERACT 2003)*. 2003. Zürich, Switzerland.

33. John, B.E., Bass, L., Kazman, R. and Chen, E., *Identifying Gaps between HCI, Software Engineering and Design, and Boundary Objects to Bridge Them*. in *CHI 2004*. 2004. Vienna, Austria.

34. Kazman, R., Bass, L. and Bosch, J., *Between Software Engineering and Human-computer Interaction, Workshop at ICSE 2003*. in *ICSE 2003*. 2003. Portland, Oregon.

35. Kazman, R., and Bass, L., (eds.), *Special Issue on Bridging the Process and Practice Gaps between software Engineering and Human computer Interaction.* Software Process - Improvement and Practice, 2003.

36. Seffah, A. and Metzker, E., *The Obstacles and Myths of Usability and Software Engineering.* Communications of the ACM, 2004. 41(12): p. 71-76.

37. Sjoberg, D.I.K., Dyba, T. and Jorgensen, E. *The Future of Empirical Methods in Software Engineering Research*. in *Future of Software Engineering, 2007. FOSE '07*. 2007.

38. WoSQ. *Fifth Workshop on Software Quality, at ICSE 07*. 2007 [cited 2009-04-13]; Available from: http://attend.it.uts.edu.au/icse2007/.

39. 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.

40. Boehm, B.W., *A spiral model of software development and enhancement.* Computer, 1988. 21(5): p. 61-72.

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

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

43. Martin, D., Rooksby, J., Rouncefield, M. and 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.

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

45. Fenton, N.E. and Pfleeger, S.L., *Software Metrics: A Rigorous and Practical Approach*. 1998, Boston MA: PWS Publishing Co. 656.

46. DeMarco, T., *Mad About Measurement*, in *Why Does Software Cost So Much?* 1995, Dorset House Publishing Company: New York. p. 13-44.
47. Basili, V.R., Caldiera, G. and Rombach, H.D., *Measurement*, in *Encyclopedia of software engineering*, J.J. Marciniak, Editor. 1994, John Wiley & Sons: New York. p. 646-661.
48. Iversen, J. and Kautz, K., *Pinciples of Metrics Implementation*, in *Improving Software Organisations: From Principles to Practice*, L. Mathiassen, J. Pries-Heje, and O. Ngwenyama, Editors. 2001, Addison-Wesley: New York. p. 387-305.
49. Dittrich, Y., Rönkkö, K., Eriksson, J., Hansson, C., and Lindeberg, O., *Co-operative Method Development: Combining qualitative empirical research with method, technique and process improvement.* Journal of Empirical Software Engineering, 2007.
50. U-ODD. *Use-Oriented Design and Development*. 2008 [cited 2008-06-09]; Available from: http://www.bth.se/tek/u-odd.
51. BESQ. *Blekinge - Engineering Software Qualities*. 2006 [cited 2009-04-05]; Available from: www.bth.se/besq.
52. Seaman, C.B., *Qualitative methods in empirical studies of software engineering.* Software Engineering, IEEE Transactions on, 1999. 25(4): p. 557.
53. UIQ Technology, *UIQ Website*. 2008: www.uiq.com/aboutus.
54. UIQ Technology. *The UIQ Open Platform*. 2008 [cited 2008-06-17]; Available from: http://uiq.com/openuiqplatform.html.
55. 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.
56. Dittrich, Y., Rönkkö, K., Lindeberg, O., Eriksson, J., and Hansson, C., *Co-Operative Method Development revisited.* SIGSOFT Softw. Eng. Notes, 2005. **30**(4): p. 1-3.
57. Rönkkö, K., *Making Methods Work in Software Engineering: Method Deployment as a Social achievement*, in *School of Engineering*. 2005, Blekinge Institute of Technology: Ronneby.
58. Mumford, E., *Advice for an action researcher.* Information Technology and People, 2001. 14(1): p. 12-27.
59. Robson, C., *Real World Research*. 2nd ed. 2002, Oxford: Blackwell Publishing. 599.
60. Eriksson, J., *Supporting the Cooperative Design Process of End-User Tailoring*, in *School of Engineering - Dept. of Interaction and System Design*. 2008, Blekinge Institute of Technology: Ronneby, Sweden. p. 258.
61. 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.

62. Glaser, B.G. and Strauss, A.L., *The discovery of grounded theory : strategies for qualitative research*. 1967: Aldine Transaction. 271.

63. Strauss, A.L. and Corbin, J., *Basics of Qualitative research: Techniques and Procedures for Developing Grounded Theory*. 2nd ed. 1998, Thousand Oaks, California: Sage Publications Limited.

64. QSR International. *NVivo 8*. 2008 [cited 2008-12-05]; Available from: http://www.qsrinternational.com/products_nvivo.aspx.

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

66. Blomberg, J., Burrell, M. and Guest, G., *An ethnographic approach to design*, in *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*. 2002, Lawrence Erlbaum Associates, Inc.: Mahwah, NJ, USA. p. 964-986.

67. Hughes, J., King, V., Rodden, T. and Andersen, H., *Moving out from the control room: ethnography in system design*. in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. 1994. Chapel Hill, North Carolina, United States: ACM.

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

69. Sanoff, H., *Special issue on participatory design.* Design Studies, 2007. 28(3): p. 213.

70. Kensing, F. and Blomberg, J., *Participatory Design: Issues and Concerns.* Computer Supported Cooperative Work (CSCW), 1998. **7**(3-4): p. 167-185.

71. Investopedia.com, *Cumulative Voting*. [cited 2009-04-15]. Available from: http://www.investopedia.com/terms/c/cumulativevoting.asp

72. Wikipedia, *Cumulative Voting*. [cited 2009-04-15]. Available from: http://en.wikipedia.org/wiki/Cumulative_Voting

73. Gordon, J.N., *Institutions as Relational Investors: A New Look at Cumulative Voting.* Columbia Law Review, 1994. 94(1): p. 69.

74. Sawyer, J. and MacRae, Jr., D., *Game Theory and Cumulative Voting in Illinois: 1902-1954.* The American Political Science Review, 1962. 56(4): p. 936 - 946.

75. Leffingwell, D. and Widrig, D., *Managing Software Requirements: A Use Case Approach*. 2nd ed. 2003: Addison Wesley.

76. Berander, P. and Wohlin, C., *Identification of key factors in software process management - a case study*. in *2003 International Symposium on Empirical Software Engineering, ISESE'03*. 2003. Rome, Italy.

77. Berander, P. and Jönsson, P., *Hierarchical Cumulative Voting (HCV) - Prioritization of Requirements in Hierarchies.* International Journal of Software Engineering & Knowledge Engineering, 2006. 16(6): p. 819.

78. Jönsson, P. and Wohlin, C., *A Study on Prioritisation of Impact Analysis Issues: A Comparison between Perspectives*. in *SERPS'05*. 2005. Mälardalen University, Västerås, Sweden.

79. Berander, P. and Wohlin, C., *Differences in Views between Development Roles in Software Process Improvement - A Quantitative Comparison*. in *8th Conference on Empirical Assessment in Software Engineering (EASE 2004)*. 2004. Edinburgh, Scotland.

80. Dumas, J. and Redish, J., *A Practical Guide to Usability Testing*. 1999: Intellect. 404.

81. Nielsen, J. and Mack, R., *Usability Inspection Methods*. 1994: John Wiley & Sons, Inc. 413.

82. Nielsen, J., *Usability Engineering*. 1993, San Diego: Academic Press. 362.

83. Whiteside, J., Bennett, J. and Holtzblatt, K., *Usability Engineering: our experience and evaluation*, in *Handbook of human computer interaction*, M. Helander, Editor. 1987, North-Holland: New York. p. 791-817.

84. Bevan, N., *International standards for HCI and usability.* Int. Jourrnal of Human-Computer Studies, 2001. 55(4): p. 533-552.

85. International Organization for Standardization, *ISO/IEC 9126-4 Software engineering - Product quality - part 4: Quality in use metrics*. 2004.

86. International Organization for Standardization, *ISO 13407:1999 Human-centred design processes for interactive systems*. 1999. p. 26.

87. Wharton, C., Rieman, J., Lewis, C. and Polson, P., *The cognitive walkthrough method: a practitioner's guide*, in *Usability inspection methods* J. Nielsen and R.L. Mack, Editors. 1994, John Wiley & Sons, Inc. p. 105-140.

88. Desurvire, H.W., *Faster, cheaper!! Are usability inspection methods as effective as empirical testing?*, in *Usability inspection methods* J. Nielsen and R.L. Mack, Editors. 1994, John Wiley & Sons, Inc. p. 173-202.

89. Been-Lirn Duh, H., Tan, G.C.B. and Hsueh-hua Chen, V. *Usability Evaluation for Mobile Device: A Comparison of Laboratory and Field Devices*. in *MobileHCI'06*. 2006. Helsinki, Finland: ACM.

90. Monrad Nielsen, C., Overgaard, M., Bach Pedersen, M., Stage, J. and Stenild, S., *It's Worth the Hassle! The Added Value of Evaluating the Usability of Mobile Systems in the Field*. in *NordiCHI 2006: Changing Roles*. 2006. Oslo, Norway: ACM.

91. Jokela, T., Iivari, N., Matero, J. and Karukka, M., *The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11*. in *Latin American conference on Human-computer interaction* 2003. Rio de Janeiro, Brazil ACM Press.

92. Sauro, J. and Kindlund, E., *A Method to Standardize Usabiilty Metrics Into a Single Score*. in *CHI 2005*. 2005. Portland, Oregon, USA: ACM Press.

93. Bevan, N., *Measuring usability as quality of use.* Software Quality Journal, 1995. 4(2): p. 115-130.

94. Kitchenham, B. and Pfleeger, S.L., *Software quality: the elusive target [special issues section]*. Software, IEEE, 1996. 13(1): p. 12.

95. Wong, B. and Jeffery, R., *Cognitive Structures of Software Evaluation: A Means-End Chain Analysis of Quality*, in *Product Focused Software Process Improvement : Third International Conference, PROFES 2001*. 2001, Springer Berlin / Heidelberg: Kaiserslautern, Germany. p. 6 - 26.

96. Frøkjær, E., Hertzum, M. and Hornbæk, K., *Measuring Usability: Are Effectiveness, Efficiency, and Satisfaction Really Correlated?* in *Conference on Human Factors in Computing Systems*. 2000. The Hague, Netherlands: ACM Press.

97. UIQ Technology, *UTUM Report: Inside Information*. [PDF file] 2006 [cited 2009-04-16]; Available from: http://www.uiq.com/files/documents/UTUM_report_Inside_Information.pdf.

98. UIQ Technology, *UIQ Technology Usability Metrics*. 2006-11-15, UIQ Technology: http://uiq.com/utum.html.

99. Symbian, *Symbian Smart Phone Show*. 2007, http://www.symbiansmartphoneshow.com/2006/: London.

100. UIQ Technology. *UTUM website*. 2006-09-13 [cited 2007-06-07]; Available from: http://uiq.com/utum.html.

101. UIQ Technology. *UTUM Report: Inside Information*. [PDF file] 2006 [cited 2009-04-16]; Available from: http://www.uiq.com/files/documents/UTUM_report_Inside_Information.pdf.

102. BTH. *UIQ Technology Usability test*. 2008 [cited 2009-04-16]; Available from: http://www.youtube.com/watch?v=5IjIRlVwgeo.

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

104. CQM, *A special issue on Kano's Methods for Understanding Customer-defined Quality.* Center for Quality of Management Journal, 1993. **2**(4): p. 37.

105. Winter, J., Rönkkö, K., Ahlberg, M. and Hotchkiss, J., *Meeting Organisational Needs and Quality Assurance through Balancing Agile & Formal Usability Testing Results*. in *CEE-SET 2008*. 2008. Brno.

106. Denman, G., *The Structured Data Summary (SDS)*. 2008.

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

108. The Agile Alliance, *Principles of Agile Software*. 2001 [cited 2008-06-12]; Available from: http://www.agilemanifesto.org/principles.html.

109. Cockburn, A., *Agile Software Development*. The Agile Software Development Series, ed. A. Cockburn and J. Highsmith. 2002, Boston: Addison-Wesley.

110. Boehm, B., *Get Ready for Agile Methods, with Care.* Computer, 2002. 35(1): p. 64-69.

111. Rönkkö, K., Winter, J. and Hellman, M., *Reporting User Experience through Usability within the Telecommunications Industry*, in *CHASE Workshop, ICSE 2008*. 2008: Leipzig, Germany.

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

113. NN/g. *Nielsen Norman group: User Expereince - Our Definition*. 2008 [cited 2008-06-09]; Available from: http://www.nngroup.com/about/userexperience.html.

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

115. COST294. *COST Action 294: MAUSE*. [cited 2008-06-09]; Available from: http://cost294.org/.

116. COST. *European Cooperation in the field of Scientific and Technical Research*. 2008 [cited 2008-06-09]; Available from: http://www.cost.esf.org/.

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

118. CHI2008. *Now Let's Do It in Practice: User Experience Evaluation Models in Product Development*. 2007 [cited 2009-04-16]; Available from: http://www.cs.tut.fi/ihte/CHI08_workshop/introduction.shtml.

119. COST294, *COST294 Newsletter, April 2008*. 2008: [cited 2009-04-16] http://cost294.org/upload/521.pdf.

120. COST294 and SIG1. *Responses to the SIG-UX questionnaire*. 2008 [cited 2009-04-16]; Available from: http://cost294.org/sig-ux-results.html.

121. Saffer, D., *Designing for interaction*. 2007, Berkely, CA: New Riders. 232.

122. Swallow, D., Blythe, M. and Wright, P., *Grounding experience: relating theory and method to evaluate the user experience of smartphones*, in *Proceedings of the 2005 annual conference on European association of cognitive ergonomics*. 2005, University of Athens: Chania, Greece.

123. Brooke, J. *SUS - A quick and dirty usability scale*. [cited 12 Jan 2007].

Appendices

Appendix A: The UTUM test

In the following, we use a series of tests that were performed in two countries in 2007 to exemplify the test, the test procedure, and the way the test data can be used. The testing was performed by two test experts in two countries, and three phones were included in this round of tests: one UIQ phone, one other "Smart phone" of a competing brand, and a popular consumer phone.

The test itself does not take place in a laboratory environment, but should preferably take place in a familiar environment for the person who performs the test, in order that he or she should feel comfortable. If this is not possible it should take place in an environment that is as neutral as possible. To avoid creating an atmosphere of stress, the test leader books one hour with the tester, although the test itself takes around twenty minutes to perform.

Choosing use cases, devices and test participants

Choosing the use cases. The use cases in the test cycle detailed here were decided by one of the companies participating in the study, and were chosen from the 20 most important use cases for a certain model of mobile phone. Otherwise, the choice of use cases can for example be based on the current tester's preferred mobile phone use, on a request from a customer, or on the basis of answers to questionnaires completed during interviews performed with many users, where respondents have ranked the applications that they considered most important and ranked different aspects of the different features. The use cases in this particular test cycle are regarded as being the beginning of a baseline of usability data that can be drawn from and compared to in the future.

The six use cases were:

- UC1. Receive and answer an incoming call
- UC2. Save the incoming call as a new contact – "Mårten"
- UC3. Set an alarm for 8 o'clock tomorrow morning
- UC4. Read an incoming SMS and reply with "I'm fine"
- UC5. Make a phone call to Mårten (0708570XXX)
- UC6. Create a new SMS – "Hi meet at 5" and send to Mårten (0708570XXX)

Choosing the participants. The test group consisted of 24 testers from Sweden, and 24 testers from England, split into 3 age groups:

- 17 – 24
- 25 – 34
- 35+

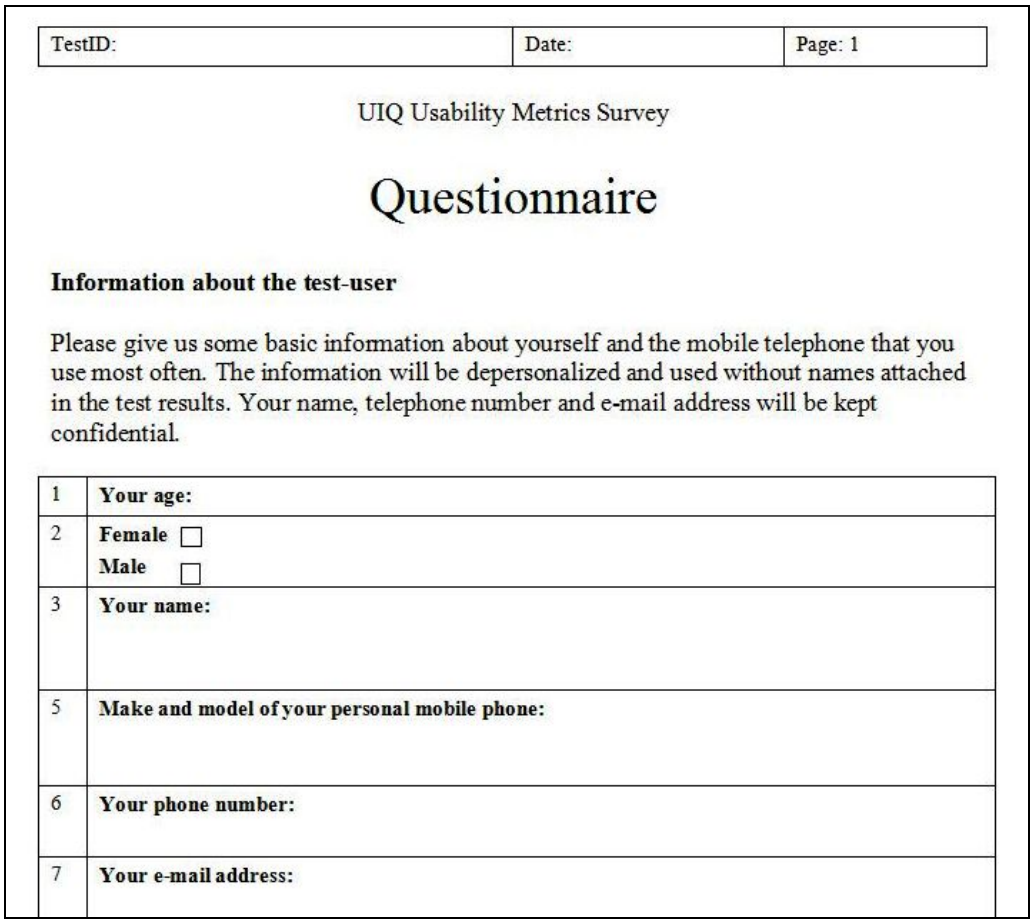
Each age group consisted of 8 females and 8 males. The size of the group was in order to get results from a wide range of testers to obtain general views, and

to enable comparisons between age groups, cultures and genders. They were drawn from a large database of mobile phone users who have expressed an interest in being testers, and who may or may not have been testers in previous projects.

The test situation

Welcoming the participant. The test leader welcomes the test participant, tries to set them at their ease, and explains the purpose and process of the test. It is explained that it is the device that is being tested, and not the user's performance, and that the tester should tell the test leader when to start the stopwatch for timing the use case, and when the use case is complete.

Recording test participant data. The tester is asked to fill in the participant data form recording personal details and usage patterns (see Figure A.1).



TestID:	Date:	Page: 1

UIQ Usability Metrics Survey

Questionnaire

Information about the test-user

Please give us some basic information about yourself and the mobile telephone that you use most often. The information will be depersonalized and used without names attached in the test results. Your name, telephone number and e-mail address will be kept confidential.

1	**Your age:**
2	**Female** ☐ **Male** ☐
3	**Your name:**
5	**Make and model of your personal mobile phone:**
6	**Your phone number:**
7	**Your e-mail address:**

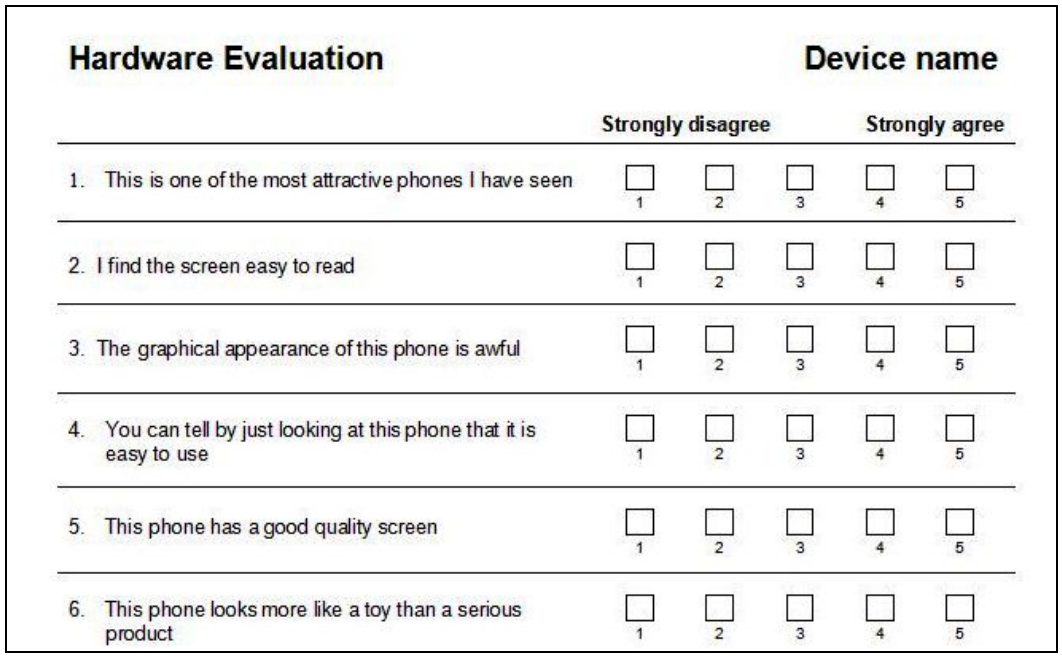
Figure A.1. The participant data form

This records name, age, gender, previous telephone use, and other data that can have an effect on the result of the test, such as which applications they find most important or useful. In some circumstances, this data can also be used to choose use cases for testing, based on the tester's use patterns.

Getting acquainted with the device. In the next step, the tester is introduced to the phone or phones to be tested. If several devices are to be tested, all of the

use cases are performed on one device before moving on to the next phone. The tester is given a few minutes to get acquainted with the device, to get a feeling for the look and feel of the phone.

Hardware evaluation. After a few minutes, the tester is asked to fill in a Hardware Evaluation (HWE), which is a questionnaire based on the System Usability Scale (SUS) [103] regarding attitudes to the look and feel of a specific device (see Figure A.2). The questions are based on Likert style questions. The Likert scale is a widely used summated rating that has the advantage of being easy to develop and use. People often enjoy completing this type of scale, which can be important, as they are likely to give considered answers and be more prepared to participate in this than in a test that they perceive as boring ([59] p. 293).



Hardware Evaluation **Device name**

	Strongly disagree				Strongly agree
1. This is one of the most attractive phones I have seen	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
2. I find the screen easy to read	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
3. The graphical appearance of this phone is awful	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
4. You can tell by just looking at this phone that it is easy to use	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
5. This phone has a good quality screen	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
6. This phone looks more like a toy than a serious product	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

Figure A.2. The hardware evaluation form

Performing the use cases. In the performance test, the users performed the use cases presented previously. The test leader observes the user whilst the testing is proceeding. He or she records the time taken to execute the individual tasks, observes user hesitation and divergences from a natural flow of user interaction, notes errors, and counts the number of clicks needed to complete the task. The test data is recorded in a form where the test leader can make notes of his or her observations during the use case performance (see Figure A.3). The test leader ranks the results of the use case on a scale between 0 – 4, where 4 is the best result. This is a judgement based on the experience and judgement of the test leader, and allows the test leader to give a result that is not simply based on the time taken to perform the use case, but also on the flow of events, and events that may have affected the completion of the use case.

TestID:	Date:	Test leader:

Use case ID:	Softkey style:	Pen style:	Softkey:	Combination:

Use case ID:	Softkey style:	Pen style:	Softkey:	Combination:

Figure A.3. The test leader's record

Task effectiveness evaluation. After performing each use case, the tester completes a Task Effectiveness Evaluation, a shortened SUS questionnaire [103] concerning the phone in relation to the specific use case performed (see Figure A.4).

This process is repeated for each use case to be performed. Between use cases, there is time to discuss what happened, and to explain why things happened the way they did. The test leader can discuss things that were noticed during the test and see whether his or her impressions were correct, and make notes of comments and observations.

Task Effectiveness Evaluation

	Strongly disagree				Strongly agree
1. This phone works well for accomplishing this task.	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
2. I am disappointed with the way this phone accomplishes this task.	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
3. This task is easy to accomplish when using this phone.	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

Figure A.4. The Task Effectiveness Evaluation

Attitudinal evaluation. The System Usability Scale. The final step in the test is an attitudinal metric that represents the user's subjective impressions of how easy the interface is to use, and this is found through the use of the System Usability Scale (SUS) [103], which expresses the tester's opinion of the phone as a whole (see Figure A.5). The statements in the original SUS questionnaire are modified slightly, where the main difference is the replacement of the word "system" with the word "phone" to reflect the fact that a handheld device is being tested, rather than a system.

System Usability Scale

	Strongly disagree				Strongly agree
1. I would like to use this phone frequently.	1	2	3	4	5
2. I found this phone unnecessarily complex.	1	2	3	4	5
3. This phone was easy to use.	1	2	3	4	5
4. I would need the support of a technical person to be able to use this phone.	1	2	3	4	5

Figure A.5. The System Usability Scale

The SUS is a "quick and dirty" usability scale based on ISO 9241:11. It is suitable here since it is simple to use, is a well established tool in the usability community, and because of its basis in the ISO standard. SUS is a simple ten-item Likert scale that provides an overview of subjective assessments of usability. The SUS results in a number that expresses a measure of the overall usability of the system as a whole (see [123] for a detailed description). It is a record of the tester's immediate response to the test situation, and is generally used after the user has had a chance to use the system being evaluated, but before any debriefing or discussion of the test. The tester fills in the SUS form together with the test leader, giving an opportunity to discuss issues that arose during the test situation.

A large number of metrics are calculated from the data obtained in the testing procedure, and these are presented in the next section of this chapter.

Metrics and observational data

The Usability metrics and their uses. The results of the testing, in the form of data concerning the individual testers, the results measured in the tests, and the test leader's observations, are stored in a spreadsheet. A selection of this data can be used to make different presentations of the results to various stakeholders.

The Task Effectiveness Metric is determined by looking at each use case and determining how well the telephone supports the user in carrying out each task. It is in the form of a response to the statement "This telephone provides an effective way to complete the given task". It is based on the test leader's judgement of how well the use case was performed, recorded in the test leader's record (see Figure A.3) and the answers to the Task Effectiveness Evaluation (see Figure A.4). The user's response is calculated as an average of seven values, of which six come from the Task Effectiveness Evaluation form, whilst the seventh is the test leader's evaluation of task effectiveness, which is tempered by the observations that have been made during the test.

The Task Efficiency Metric is a response to the statement "This telephone is efficient for accomplishing the given task". It is calculated by looking at the distribution of times taken for each user to complete each use case. For example, in this case, where three devices were tested, the distribution of completion times for each device is plotted in a graph, giving three separate curves, one for each device (and a curve showing the average of all devices). The distribution of completion times is used to calculate an average value for each device per use case (see Figure A.6).

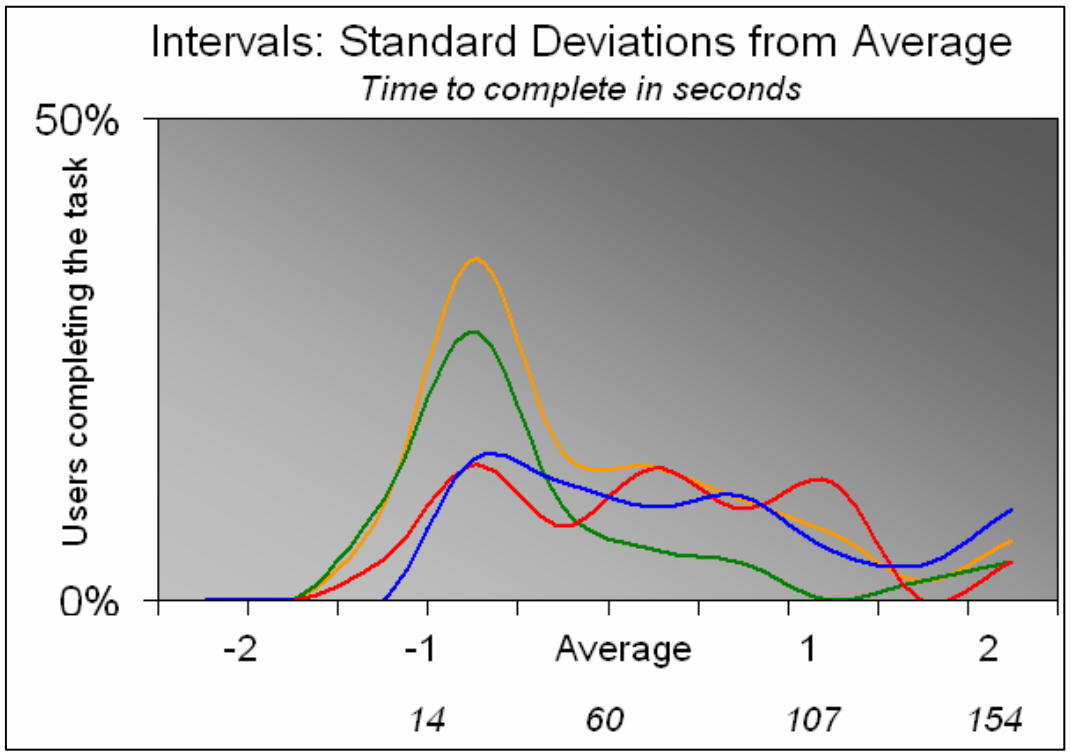


Figure A.6. Distribution of completion times

The User Satisfaction Metric, is calculated as an average score for the ten answers in the SUS (see Figure A.5), and is a composite response to the statement "This telephone is easy to use". The mechanics of this operation are described more closely in ([103]).

These metrics can be combined in diagrams to illustrate different aspects of the testing.

To illustrate the Effectiveness-Efficiency metric (this diagram is also called the Task Effectiveness Metric, see Figure A.7), the average Task Effectiveness metric for all of the tests of a single use case, is contrasted with Task Efficiency, the average of the performance efficiency metric for all of the tests of the same use case. Increased usability for the specific task is shown as a move towards the upper right of the quadrant.

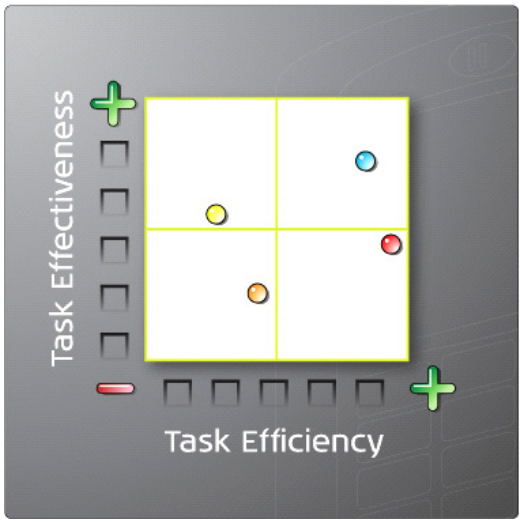


Figure A.7. The Effectiveness-Efficiency Metric

To illustrate the Satisfaction-Efficiency Metric, the User Satisfaction Metric and the average Task Efficiency Metric for all of the use cases and all test cases are contrasted (see Figure A.8). Increased usability is shown as movement towards the upper right of the quadrant. The Satisfaction-Efficiency Metric is referred to as the Total UTUM, and is regarded as a useful illustration of total usability.

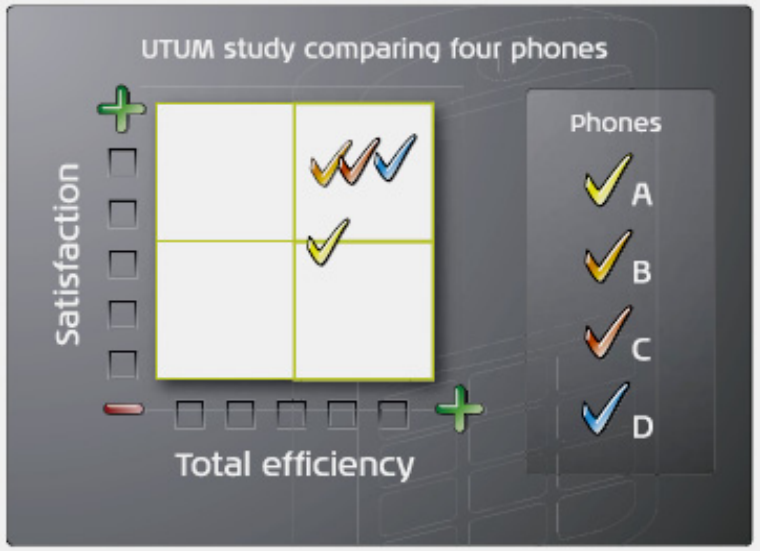


Figure A.8. Total UTUM [101]

The results shown above are calculated from the metrics, and the figures are available those who can interpret them, but no numerical values are shown in the diagrams. This is a result of discussions with stakeholders, when the test

leaders found that the numbers often had a tendency to get in the way of communicating the message. This was found to be an effective way of presenting results.

The diagrams can e.g. show: Total UTUM for a complete test series; Total UTUM by gender; Total UTUM by age groups; Total UTUM per tester, and; Efficiency per use case. This is shown on a simple scale showing an average for each separate device. This is just a selection of all of the possible ways of using the data to illustrate different facets of the test, and the presentations can be adapted to different stakeholders' needs.

Appendix B: Data table, Chapter 6

	D	PO	D	D	PO	D	PO	D	PO
	UI Designer	Product planning	System Design	Other	Other	UI Designer	System Design	System Design	System Design
Role				Usability	CTO Office				
Other Role									
Method 1	10		20					20	10
Rank	4		1					1	3
TL Help	No		Yes	Yes		Yes		No	
Method 2	15							5	13
Rank	2							9	2
TL Help	No		No	Yes		Yes		Yes	
Method 3			20	10		50		10	7
Rank			1	7		1		4	6
TL Help			Yes	Yes		No		No	
Method 4		25		11		5		8	5
Rank		2		6		5		6	7
TL Help		Yes		No		No		No	
Method 5		20		12		20		12	10
Rank		3		5		2		3	3
TL Help		Yes		No		No		No	
Method 6			20	14	15	15	50	16	30
Rank			1	3	2	3	1	2	1
TL Help			Yes	No		No		No	
Method 7		15	5	9				5	5
Rank		4	6	8				9	7
TL Help		Yes	Yes	Yes		Yes		Yes	
Method 8	15			13	70		25	6	5
Rank	2			4	1		2	8	7
TL Help	No			No		Yes		No	
Method 9	50	10	20	16	10	10	15	8	5
Rank	1	5	1	1	3	4	3	6	7
TL Help	Yes	Yes	Yes	Yes	Yes	Yes		No	
Method 10	10	30	15	15	5		10	10	10
Rank	4	1	5	2	4		4	4	3
TL Help	Yes	Yes	Yes	Yes	Yes	Yes		Yes	

Table B.1. Answers for all respondents