Abstract

The objective of this thesis is to obtain knowledge regarding how effective user centred evaluation methods are and how user centred evaluations are conducted by IT professionals. This will be achieved by exploring user centred evaluation in experimental and practical settings. The knowledge gained in these studies should inspire suggestions for further research and suggestions for improvements on the user centred evaluation activity.

Two experimental studies were conducted. One compares the results from using three user centred evaluation methods, and the other examines two factors while conducting heuristic evaluation. The results show that the think-aloud evaluation method was the most effective method in finding realistic usability problems of the three methods. The number of critical problems found during think-aloud evaluation increases, if heuristic evaluation is conducted prior to the think-aloud evaluations.

Further, two studies of user centred evaluation in practical settings were performed. The IT professionals participating in those studies were using the software development process Scrum to plan their work. The results show that user centred evaluation is infrequently conducted in Scrum projects, compared to testing activities like acceptance testing. The main type of evaluation is qualitative. Few participants measure user performance or use surveys to gather quantitative results on the usability and the user experience. IT professionals get feedback from users in an informal way and gather informal feedback from peers. Many participants use a mixture of methods for gathering feedback on their work.

The outcome of this thesis shows that IT professionals should be encouraged to include users whenever possible when evaluating software, for example by using the think-aloud method. Using heuristic evaluation prior to conducting think-aloud evaluations is also recommended. In addition, IT professionals are encouraged to evaluate their software in an informal way frequently, rather than waiting for the right time to conduct a thorough quantitative evaluation.

To advance this field further, researchers who want to improve the evaluation activity for the IT professionals should study how user centred evaluation methods could be combined in an efficient way and how the use of qualitative evaluation methods could be made more effective.
Acknowledgements

All my best thanks go to my husband Guðmundur, for being so positive during the whole period of the PhD studies. I could never have done this without your full support. You did enjoy taking care of the kids while I was away and you were caring while I was writing this thesis.

Secondly, I want to thank my supervisor Jan Gulliksen, for encouraging and constructive feedback and discussions. Your advices, the discussions we have had and our co-operation has been very valuable. I also appreciate your and your wife’s great hospitality. It was always so friendly to come to your place and dine with you and your family.

Thirdly, I want to thank my co-supervisor and friend Åsa Cajander at Uppsala University. I really admire your positive attitude. We have had great discussions, outstanding co-operation and you have given me so many good advices. Many thanks to the whole HCI research group at Uppsala University for being very welcoming during my visits.

Fourth, I want to thank my co-supervisor Ann Lantz for giving me many issues to think about during my studies. In addition, I want to thank the HCI group at KTH, especially Elina Eriksson, Åke Walldius and Yngve Sundblad for their good advices. Moreover, I want to thank Niklas Hallberg for his outstanding support and constructive comments.

I want to thank my colleagues at Reykjavik University for being so supportive, especially my dean and friend Björn Þór Jónsson, for being very encouraging and positive regarding my studies. Moreover, I want to thank Anna Ingólfsdóttir and Yngvi Björnsson, for their good advices. Additionally, my special thanks go to Luca Aceto, who did read one of the final versions of this thesis and gave me valuable comments.

I want to thank my fellow authors for their contribution in this work. Moreover, I thank the COST-294 action and the COST- IC0904 for sponsoring short term scientific missions for me at Uppsala University and at the Royal Institute of Technology. Finally, many thanks go to my friends and family that have all been very supportive and encouraging, especially to Janet Read and Borghildur Jóhannsdóttir for their valuable support.

It is a pleasure for me to experience how all this people have been willing to give me of their time and energy. Thereby you have all helped in making the PhD studies a positive experience for me.
Preface

User centred evaluation, the subject of this thesis, has been dear to me for over 20 years. During my Master studies in Computer Science at Copenhagen University I discovered the field of Human-Computer Interaction, when the first course was given at the university in the area. It really made a big difference in my enthusiasm for the studies to consider the users’ perspective in software development. I wanted to examine how IT professionals design user interfaces and how they can evaluate their designs. Additionally, I was very interested in examining how effective the evaluation methods were for IT professionals. These issues continue to motivate me today through my teaching and research.

Many roles of IT professionals in Iceland interested me after my Master studies. I chose the role of a usability expert at a company buying software for internal use. I evaluated software with real users participating, and described the usability requirements for the software. I worked there for two years, but decided that I would have more impact as a usability expert at a software company, taking part in several software development projects at the same time. Moreover, I took the role of a project manager, a tester and a trainer for the prospective international users. After being at this software company for five years working as an IT professional, I got a permanent position at Reykjavik University as an assistant professor, teaching and researching in the field of Human-Computer Interaction. I wanted to educate the majority of Icelandic computer science students in the area and thereby have more impact on software development in Iceland, than being an IT professional at one software company. That has been my occupation for over ten years. During these years, I have been teaching many courses in Human-Computer Interaction. Moreover, I have been researching user centred evaluation to be able to suggest efficient ways of evaluating interactive software both to my students and IT professionals.

All this time I have found user centred evaluation extremely important and interesting. Through the years there have been many interesting research questions to examine both in experimental and practical settings. I am convinced that I will have as many interesting questions to examine in the future.
The Papers Included in the Thesis

This summary is based on the following papers, which are referred to in the text by Roman numerals. Full version of each of the four papers is included in the thesis.

**Paper I**  
**Prediction of usability: Comparing method combinations.**

Authors Frøkjær, E. and Larusdottir, M.K.
Year 1999
Publication *In proceedings of the conference: Managing information technology resources in organizations in the next millennium*, Hershey, USA. 248 – 257.

My contribution The study was my idea and was conducted during my Master studies in Copenhagen University. I made the research plan, conducted the experiment and analysed the data under the supervision of my co-author. I wrote a project report which the paper is based on. I was active in writing the paper. The authors were listed in alphabetic order.

**Paper II**  
**Heuristic evaluation: Comparing ways of finding and reporting usability problems.**

Authors Hvannberg, E.P., Law, E.L. and Larusdottir, M.K.
Year 2007

My contribution I supervised the students that gathered the data from the evaluations and took active part in planning the experiment and analysing the data. I was active in writing the paper.
Paper III  The focus on usability in testing practices in industry.

Authors  Larusdottir, M.K., Bjarnadottir, E. and Gulliksen, J.

Year  2010

Publication  *In proceedings from the Human-Computer Interaction symposium at the World Computer Congress*, Brisbane, Australia. 98 – 109.

My contribution  I planned the study together with one of my co-authors, Emma Bjarnadóttir, which was my student at that time. I co-operated with her in designing the questionnaire and analysing the results from the questionnaire. I analysed the results from the interviews. I was the leading author for writing the paper.

Paper IV  Informal feedback rather than performance measurements - User centred evaluation in Scrum projects.

Authors  Larusdottir, M.K., Cajander, Å. and Gulliksen, J.

Year  2012

Publication  *Behaviour and Information Technology*, submitted on the 30th of April 2012, 36 pages.

My contribution  I planned the study together with my co-authors. I was the interviewer in most of the interviews. I and the second author coded and analysed the material together. I interpreted the results together with my co-authors. I was the leading author for writing the paper.

A list of my other publication not included in this thesis is given in Appendix I.
# My Co-authors

<table>
<thead>
<tr>
<th>Name</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emma Rún Bjarnadóttir</td>
<td>Test Engineer, Sabre Airline Solutions, Iceland.</td>
</tr>
<tr>
<td>Åsa Cajander</td>
<td>Assistant Professor, Division of Human-Computer Interaction, Department of Information Technology, Uppsala University, Sweden.</td>
</tr>
<tr>
<td>Erik Frøkjær</td>
<td>Associate Professor, Department of Computer Science, Copenhagen University, Denmark.</td>
</tr>
<tr>
<td>Jan Gulliksen</td>
<td>Professor of Human-Computer Interaction, Dean of the School of Computer Science and Communication, Royal Institute of Technology (KTH), Sweden.</td>
</tr>
<tr>
<td>Ebba Þóra Hvannberg</td>
<td>Professor, Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland, Iceland.</td>
</tr>
<tr>
<td>Effie Law</td>
<td>Reader, Department of Computer Science, University of Leicester, England and visiting Senior Research Assistant at Swiss Federal Institute of Technology (ETH), Switzerland.</td>
</tr>
</tbody>
</table>
# Table of Contents:

1 INTRODUCTION .................................................................................................................................................. 1

2 THE RESEARCH FIELD AND THE RESEARCH TOPIC ............ 5

2.1 THE RESEARCH FIELD – HUMAN-COMPUTER INTERACTION .......... 5
2.2 THE RESEARCH TOPIC - USER CENTRED EVALUATION ............ 6
   2.2.1 Usability ................................................................................................................................. 7
   2.2.2 User Experience ...................................................................................................................... 11
2.3 REMARKS ON THE CONCEPTS ................................................................. 14

3 EXAMINING USER CENTRED EVALUATION ......................15

3.1 THE CONTEXT OF USER CENTRED EVALUATION ......................15
3.2 THE UCE MODEL .......................................................................................................................... 17
3.3 RESEARCHING EVALUATION IN EXPERIMENTAL SETTINGS ........21
3.4 RESEARCHING EVALUATION IN PRACTICAL SETTINGS ............ 23
3.5 OVERVIEW OF THE UCE FACTORS STUDIED ...................... 24

4 EXAMINING EVALUATION IN EXPERIMENTAL SETTINGS ......25

4.1 RELATED WORK ................................................................................................................................. 25
   4.1.1 User Centred Evaluation Methods ............................................................... 26
   4.1.2 The Effectiveness of Using Evaluation Methods .................................... 30
   4.1.3 Supporting Material in Heuristic Evaluation ......................................... 31
   4.1.4 Data Gathering Equipment while Conducting Evaluation ................. 32
   4.1.5 Evaluator’s Knowledge of Evaluation Methods ........................................ 32
4.2 DATA GATHERING AND ANALYSIS ................................................. 34
   4.2.1 Gathering Usability Problem Lists ............................................................... 34
   4.2.2 Analysing the Effectiveness of the Evaluations .................................. 35
4.3 RESULTS .......................................................................................................................... 37
   4.3.1 Paper I: Prediction of Usability ................................................................. 37
   4.3.2 Paper II: Heuristic Evaluation ................................................................. 38
4.4 DISCUSSION ......................................................................................................................... 39
4.5 LESSONS LEARNED .......................................................................................................... 41
5 EXAMINING EVALUATION IN PRACTICAL SETTINGS ..........43

5.1 BACKGROUND ON SOFTWARE DEVELOPMENT ..................44
  5.1.1 Software Development Processes ................................44
  5.1.2 The Agile Software Development Process Scrum ...............47
  5.1.3 Human-Centred Design Approach ..............................49

5.2 RELATED WORK ON THE UCE FACTORS .......................50
  5.2.1 Evaluation Methods Used ....................................50
  5.2.2 The Context of Evaluation ..................................51
  5.2.3 The Purpose of Conducting Evaluation ........................52
  5.2.4 The Evaluators and Users ..................................54

5.3 DATA GATHERING AND ANALYSIS IN PRACTICAL SETTINGS ......55
  5.3.1 Questionnaires ..............................................55
  5.3.2 Semi Structured Interviews ..................................55

5.4 RESULTS ...................................................................57
  5.4.1 Paper III: The Focus on Usability Testing .....................57
  5.4.2 Paper IV: Evaluation in Scrum Projects .......................58

5.5 DISCUSSION .........................................................59

5.6 LESSONS LEARNED ................................................61

6 CONTRIBUTION ..........................................................63

7 FUTURE WORK ................................................................65

8 CONCLUSION ..................................................................69

9 REFERENCES ..................................................................71

10 APPENDIX I: OTHER PUBLICATION ..............................79
A vast majority of people in the western world use various types of software\(^1\) on a daily basis for achieving their goals. A typical scenario would be that a person uses her smart phone for the wake-up call in the morning, reads the news on her laptop computer during breakfast and reads e-mails on the train to work using the phone. At the workplace she gets messages from the software in the coffee machine on what type of coffee she chose before starting work and during her daily work hours she uses different types of software on a big computer screen. More than 90\% of the Icelandic population use software daily according to a recent study (Statistics Iceland 2011) and in Sweden 83\% of the salaried professionals use software on their computers more than three hours a day (Unionen 2011).

The usage of software needs to be easy for the users, especially when the software is new to them. If it is hard to use, the users could quit doing what they want to do; it could take too long time for them to achieve their goals and they could get irritated. In the worst cases the users will never use the software again and will find other ways of achieving their goals. It is hard for IT professionals\(^2\) developing new software to imagine how easy or hard it will be for the actual users to use the software (Bak \textit{et al.} 2008). To estimate the usability of new software, the IT professionals need to evaluate how easy it will be for the prospective users to use it. To this end, IT professionals conduct user centred evaluation.

The goal of the user centred evaluation activity is to gather feedback on the IT professional’s work from the user’s perspective (ISO 2010). The evaluation activity needs to be planned, conducted, and the results need to be analysed and reported (ISO 2010). Evaluation can be conducted to gather feedback on the description of the context, in which the software will be used, the requirements from the user’s perspective and on the user interface design. When the feedback has been gathered in the evaluation, the positive aspects and the flaws are described and it is decided what

---

\(^{1}\) The term software is used for set of computer programs, procedures and associated documentation and data according to ISO 12207 (2008).

\(^{2}\) The term IT professional is used in this thesis to encompass all those working actively on software development.
action to take in the particular cases in order to extend the usability and the user experience\(^3\) of the software.

There are various types of methods used to conduct user centred evaluation. Experts in user interface design could be asked to inspect the interface and list all the problems that the users could have while using the software. Another possibility is to observe real users using the new software and gather information on what happens. While observing users, the problems that the users have can be registered. In addition, data can be gathered on whether they could solve their tasks and logs can be made of how long time it took to solve each task. Furthermore, it is possible to ask the users how satisfied they were after using the software and what the experience was while using it.

Researchers have studied evaluation methods for the last 20 years in order to be able to guide IT professionals while choosing an evaluation method (Woolrych \textit{et al.} 2011). Researchers have, for example, compared the results of using more than one method for evaluating particular software in experimental settings, like is done in paper I in this thesis. Information on problems that the users have, or might have, is gathered during the evaluation in that type of research studies. Afterwards the numbers of problems found by using each method are compared, like is done in papers I and II in this thesis. The evaluation method that was used to find the highest number of problems is generally declared as the most effective method. In parallel to this, an important aspect to study is what method was used to find the highest number of severe problems. Another possibility is to study if a particular supporting material used during the evaluation changes the effect of the evaluation, like is done in paper II in this thesis. The motivation for these research studies is that researchers want to provide the IT professionals with new knowledge so that they can evaluate software effectively and thereby make their software more usable.

Some researchers have focused on examining user centred evaluation in practical settings, like is done in papers III and IV in this thesis. They want to study, for example, how much IT professionals actually evaluate during their software development like done in paper III in this thesis and how the various methods fit into the working environment of the IT professionals like done in paper IV in this thesis. Some evaluation

\(^3\) These concepts will be defined in detail in Chapter 2.
methods might fit better than others when the IT professionals use a formal process to manage the software projects and other evaluation methods might fit better for more informal processes like the agile development processes.

The agile process Scrum has gained popularity in the software industry in the Nordic countries in recent years. One third of IT professionals in Iceland used this process in 2009 (Larusdottir et al. 2009). In Scrum, self-organizing and well compounded teams are emphasized, typically with six to eight interdisciplinary team members. In Scrum, the projects are split up in two to four week long iterations called sprints. At the end of each sprint, a potential shippable product is delivered to the customer, meaning that it should be functioning for the users. Because of the short sprints and the emphasis on completing the particular part of the software during each sprint, the IT professionals do not have much time in their development for involving users and for conducting user centred evaluation. It is important to study how evaluation is conducted to be able to suggest new ways of conducting user centred evaluation to the IT professionals using the Scrum process.

**The Objective of the Research Studies**

The objective of the research studies in this thesis is to contribute new knowledge by examining the user centred evaluation activity in experimental and practical settings in the research field of Human-Computer Interaction. Particularly, studies were conducted on how user centred evaluation methods can be compared and how user centred evaluation is conducted by IT professionals developing software using the software development process Scrum. This new knowledge should provide suggestions for improving the user centred evaluation activity.

To summarize, the following two main research questions are studied in this thesis:

1. How different are the results of using various user centred evaluation methods for evaluating the same software?
2. How do IT professionals conduct user centred evaluation in the Scrum development process?
Summary of Results

The results from papers I and II in this thesis indicate that the answer to the first research question is that using the evaluation methods cognitive walkthrough, heuristic evaluation and the think-aloud method\(^4\) to evaluate the same software gives different results. In addition, the results show that the use of the think-aloud method can be recommended to IT professionals because it has proven to be effective. Combining the use of heuristic evaluation and the think-aloud method is preferable because that is even more effective, especially when finding critical problems.

The results from papers III and IV in this thesis indicate that a possible answer to the second research question is that IT professionals using the Scrum development process conduct user centred evaluation in an informal way. They often evaluate the software with real users or with user surrogates. Not many IT professionals conduct formal measurements on how easy the software is to use.

Structure of the Thesis

The research field and the research topic in this thesis are explained in Chapter 2. In Chapter 3 the context of user centred evaluation is described and the UCE model is presented for explaining the factors of user centred evaluation that can be examined in experimental and practical settings. Moreover, these two research approaches are explained. In Chapter 4, user centred evaluation is examined in experimental settings. The background material and related work is described, and the experimental studies in this thesis are summarised and discussed. In Chapter 5 user centred evaluation is studied in practical settings. The same structure is used as in Chapter 4. The thesis concludes describing the contribution, the future work and the conclusions. Full version of each of the four papers is included in the thesis.

The rationale for this structure is that it will be easier for readers either interested in studies on evaluation in experimental settings or studies on evaluation conducted in practical settings to read the thesis.

---

\(^4\) These evaluation methods will be explained in Chapter 3.
2 The Research Field and the Research Topic

In this chapter the research field of Human-Computer Interaction is described. In addition, the research topic of the thesis, user centred evaluation, will be defined. The factors evaluated in user centred evaluation, which are usability and user experience, will be defined and described. To conclude, remarks are given on the concepts described in this chapter.

2.1 The Research Field – Human-Computer Interaction

The research field of this thesis is Human-Computer Interaction (HCI). HCI is often defined by referring to the curricula for HCI from 1992 by the Association of Computing Machinery (ACM) – Special Interest Group for Computer-Human Interaction (ACM SIGCHI) (1992) as:

*Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and the study of major phenomena surrounding them (p. 5).*

The former part of the definition focuses on the activities of the IT professionals that are developing software, in the definition called interactive computing systems. The IT professionals need to include the users’ perspective in the design, the evaluation and the implementation of software. The emphasis in this thesis is on one of these three fundamental activities in the definition, namely the evaluation.

The latter part of the definition stresses the study of phenomena surrounding users, which gives it a broader scope than only including the activities of the IT professionals.

Other definitions of the HCI field can be found in the literature. In a definition by Faulkner (2000) the focus is on the interaction between the users and the software with a task oriented perspective. In this definition, the main emphasis is on how the software affects the users. Meanwhile, the process of making the software is not mentioned.

Carroll (2002) defines HCI by emphasising the term usability. The definition includes both the focus on the activities of the IT professionals as the ACM SIGCHI definition does and the focus on understanding the use of software and other technology, like in the definition from Faulkner.
When defining the HCI field it is important to include both these aspects, the activities of making software according to users’ perspectives and the understanding of how the software affects the user, like is done in the SIGCHI definition. The reason is that an understanding of both the activities of the IT professionals and an understanding of the users’ perspectives are important to be able to develop software that is easy to use.

2.2 The Research Topic - User Centred Evaluation

User centred evaluation is defined in the ISO 9241-210 (2010) standard as:

*Feedback from users is a critical source of information in human-centred design. Evaluating designs with users and improving them based on their feedback provides an effective means of minimizing the risk of a system not meeting user or organizational needs (including those requirements that are hidden or difficult to specify explicitly). Such evaluation allows preliminary design solutions to be tested against “real world” scenarios, with the results being fed back into progressively refined solutions. User-centred evaluation should also take place as part of the final acceptance of the product to confirm that requirements have been met. Feedback from users during operational use identifies long-term issues and provides input to future design (p. 6).*

*NOTE The term “user-centred” is used here to emphasize that this evaluation is made from the user’s perspective (p. 6).*

According to this definition user centred evaluation is an activity that has three main outcomes: evaluation results, conformance test results and long-term monitoring results. These results are used to improve the outcome of the other three human-centred design activities defined in the ISO 9241-210 standard (2010), which are: understand and specify the context of use, specify the user requirements and produce design solutions.

When conducting evaluation, the activity needs to be planned, the evaluations are then conducted, the outcomes are analysed and finally the results are described. These steps vary considerably depending on which method is used and in what context the evaluation is conducted, see Chapter 3 for further description of the context of user centred evaluation and the factors involved while conducting evaluation.

During user centred evaluation it is common to measure usability, user experience or both (Law 2011). The evaluation methods used are
either named usability evaluation methods or user experience evaluation methods. In the following subchapters, usability and user experience will be defined.

### 2.2.1 Usability

The term usability has several definitions where the ISO 9241-11 definition (ISO 1998) has proven to be the central one. In the following, usability according to the ISO 9241-210 (2010) standard will be defined, which is a new part in the 9241 standard series. The definition of usability in the ISO 9241-210 standard is adapted from the definition of usability in the ISO 9241-11 standard. Counting usability problems is often used as a measure of usability and will be described also.

**Usability Defined by the ISO 9241-210**

*Usability* is defined in the ISO 9241-210 (2010) as:

**Usability**: Extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. (p. 3)

*NOTE* Adapted from ISO 9241-11:1998.

The difference of the definitions of usability in these two standards is that in the 9241-11 standard (1998) a *product* can be used by the specified users, but the 9241-210 standard (2010), it is stated that a *system, product or a service* can be used. This change in the definition of usability is probably to compensate for the changes in software development that had been evolving between the two standards. The definition of an interactive system from the 9241-210 standard (ISO 2010) is:

**Interactive system**: Combination of hardware, software and/or services that receives input from, and communicates output to, users (p. 2).

A product is defined in the ISO 9241-11 standard (1998):

**Product**: Part of the equipment (software, hardware and materials) for which usability is to be specified or evaluated (p. 2).

A service is not defined in either of these ISO standards.

The definition of usability also includes three other concepts: *the user, the goal and the context of use*. The definitions for these concepts are
the same in ISO 9241-11 standard and 9241-210 standard. The definitions are (ISO 1998, 2010):

**User:** Person that interacts with the product (p. 2).

**Goal:** Intended outcome (p. 2).

**Context of use:** Users, tasks, equipment (hardware, software and materials) and the physical and social environments in which a product is used (p. 2).

There are three measureable elements in the ISO 9241-210 and the ISO 9241-11 definitions of usability: *effectiveness, efficiency and satisfaction.* Effectiveness is defined as (ISO 1998, 2010):

**Effectiveness:** Accuracy and completeness with which users achieve specified goals (p. 2).

Effectiveness is measured by assessing the extent to which a user can complete his or her goal and by measuring how accurate the outcome is. Therefore, if the user wants to insert some particular data into the software, he or she could complete that task, but make typos or select a wrong alternative from drop-down lists without knowing it. In this example, the user completed the task with low accuracy. Often the users are not even aware of the low accuracy, so they keep doing the same mistakes (Larusdottir and Armannsdottir, 2005).

*Efficiency* is defined as the resources expended in relation to effectiveness. The definition is (ISO 1998, 2010):

**Efficiency:** Resources expended in relation to the accuracy and completeness with which users achieve goals (p. 2).

It is common to consider the time it takes to complete a task as a measure of the resources expended; other measures can include keystrokes or mouse clicks.

*Satisfaction* relates to the user’s subjective judgement of how easy or hard it was to use the software. Satisfaction is defined by (ISO 1998, 2010):

**Satisfaction:** Freedom from discomfort, and positive attitudes towards the use of the product (p. 2).
Satisfaction is commonly measured by asking users to rate their satisfaction with the software by using a questionnaire, interviewing the users, or by looking at the emotional state of the users.

Usability evaluation is not defined as a term in the ISO 9241-11 standard (1998), but it is mentioned in relation to measurement. It is stated in the standard, when describing how work systems in use are specified and measured, that:

*Measures of effectiveness, efficiency and satisfaction can be used to evaluate any component of the work system (p. 7).*

The emphasis in the ISO 9241-11 standard (1998) is on measuring the three factors of usability in a quantitative manner, while conducting evaluation. Nonetheless, it is stated in the standard that (ISO 1998):

*If it is not possible to obtain objective measures of effectiveness and efficiency, subjective measures based on the user’s perception can provide an indication of effectiveness and efficiency (p.5).*

This shows that when the standard was published more focus was on quantitative measurements. Qualitative measures could be used though, to give indication of objective factors of usability, if it was not possible to obtain quantitative measurements.

**Measuring Usability by Counting Usability Problems**

A commonly used way of measuring usability is to count usability problems. A *usability problem*, or *usability defect*, is defined by Stone et al. as (2005):

*Usability defect: A difficulty in using the user interface design that affects the user’s satisfaction and the system’s effectiveness and efficiency. Usability defects can lead to confusion, error, delay or outright failure to complete some task on the part of the user. They make the user interface, and hence the system, less usable for its target users (p. 626-627).*

If the users have problems using the software, it will probably delay their work or, even worse, they may have to quit using the software before completing their tasks. If the users have many problems they will probably become irritated.

The severity of a problem can be rated by a single scale, for example, according to the need of fixing the problem, like Nielsen suggests
Another possibility is to rate the problems according to how much the problem affects the user (Molich 2000, Nielsen 1993) and the frequency with which the problem is encountered. Three categories for usability problem severity are based on descriptions from Nielsen (1993):

(a) *Low severity problem* is when the problem has small impact on few users.

(b) *Medium severity problem* is either a problem that has large impact on few users, or a problem that has little impact on many users.

(c) *High severity problem* is the one that has large impact on many users.

This categorisation for the severity of usability problems was used in paper I, because it relates directly to how much impact the problems have on users and how many users might be influenced by the particular problem.

**Usability Measures Used**

Usability measures from 180 research studies were examined by Hornbæk (2006). The results state that effectiveness and efficiency were measured in around 80% of the studies. Measuring time is the most common way for estimating the efficiency. The most common factor for measuring the effectiveness was to estimate the accuracy with which users complete tasks. That was done in around one-third of the studies. Additionally, the satisfaction was measured in around 60% of the studies. The most common factor to measure was the satisfaction with the interface.

The measures gathered in the research studies were categorized in subjective and objective measures (Hornbæk, 2006). The subjective measures were data on the perception and attitude of the users towards the software and the objective measures were data on aspects of the interaction not dependent on the users’ perspectives. In some of the research studies examined, various subjective and objective measures were mixed together. To conclude, the author recommends that special attention should be paid to whether subjective measures, objective measures or both are used when measuring usability, and special care should be taken not to confound those two types of measures (Hornbæk 2006).
2.2.2 User Experience

Some researchers have pointed out that the definition of usability is too narrow (Carroll 2004). Driven by the impression that the definitions on usability described above are too focused on task- and work-related attributes, the term user experience has gained momentum in HCI (Hassenzahl and Tractinsky 2006). User experience is defined in the ISO 9241-210 in the following way (2010):

**User Experience**: Person’s perceptions and responses resulting from the use and/or anticipated use of a product, system or service (p. 3).

There are three notes that explain the term further (ISO 2010):

*NOTE 1* User experience includes all the users’ emotions, beliefs, preferences, perceptions, physical and psychological responses, behaviours and accomplishments that occur before, during and after use (p. 3).

*NOTE 2* User experience is a consequence of brand image, presentation, functionality, system performance, interactive behaviour and assistive capabilities of the interactive system, the user's internal and physical state resulting from prior experiences, attitudes, skills and personality, and the context of use (p. 3).

*NOTE 3* Usability, when interpreted from the perspective of the users' personal goals, can include the kind of perceptual and emotional aspects typically associated with user experience. Usability criteria can be used to assess aspects of user experience (p. 3).

Researchers agree that user experience is a complex concept, including aspects like fun, pleasure, beauty and personal growth. User experience focuses on the more emotional aspects of user interactions, shifting the focus on how the users feel while and after using the software, the sensation, and the meaning as well as the value of such interactions in everyday life.

User experience is described by Hassenzahl (2003) as having pragmatic and hedonic attributes. The pragmatic attributes are task related. The main two pragmatic attributes are ease-of-use, described by effectiveness and efficiency and usefulness, described by words like clear, supporting and controllable. The main difference from the usability attributes is that users are often asked about the pragmatic attributes when evaluating user experience, so they give their subjective opinion about these attributes. The time for solving particular tasks is often used as a
measurement for efficiency when measuring usability, which is an objective measurement.

The hedonic attributes emphasize the wellbeing of the users while using the product. In the hedonic category there are three attributes the *hedonic stimulation*, *hedonic identification* and *attraction*. The *hedonic stimulation* encourages development of the user’s skills and knowledge. The *hedonic identification* describes how users identify with the software in a social context, mainly what message they want to communicate to others socially by using this product. The third hedonic attribute according to Hassenzahl (2003) is *attraction*, which usually summarizes the whole experience of the software. Words like good, bad, beautiful and ugly are used to describe the software, which is judged as a whole.

The user experience is inherently dynamic, both because of differences in circumstance during and after the use of a product and because the emotional state of the user can change quite rapidly (Hassenzahl 2008, Law et al. 2009). Thus, the user experience could be measured after interacting with a product, but also before and during the actual use (Vermeeren et al. 2010). An example of this type of study is when a task oriented software was evaluated with users both according to usability and user experience (Isleifsdottir and Larusdottir 2008). The usability of using the software was measured while observing users using the software. The user expectations about the software were measured before the user observations and the actual user experience was measured right after each evaluation using a questionnaire. The results from this study stated that there was a systematic difference between the expectations and user experience after the actual use, so both the hedonic and the pragmatic factors had a higher expectation value than the user experience value.

The changes of user experience can also be studied over longer periods. It is particularly interesting to study how user experience is affected by other factors in the context of use, for example how the user experience of using particular software changes, if a new type of equipment is used. Karapanos et al. (2009) have suggested a framework for studying the longitudinal effect of user experience. They suggest that user experience could be divided in three phases. First is the *initial phase* were the focus is on the hedonic stimulation factor of user experience, that is the
stimulation that the user gets and the learnability of the product. The second phase is the *in-corporation phase* where the pragmatic attributes, usefulness and long-term usability in daily activities become more important. Finally there is the *identification phase*, where the hedonic identification factor of user experience through the users’ personal and social experience is the dominant factor. Three forces, namely the familiarity, functional dependency and emotional attachment, are described as being the driving forces in shifting user experience across the three phases when a product is adopted (Karapanos *et al.* 2009).

The difference of usability and user experience measures has been discussed extensively, especially the difference of user satisfaction and user experience (Law 2011). One of the main discussion topics is that user satisfaction is more quantitative and user experience is more qualitative. Moreover, it has been pointed out that usability measures may hint to a particular problem and sometimes to a solution of it, whereas user experience measures are more general. This makes the usability measures more useful and persuasive for the IT professionals according to Law (2011).

It has been pointed out that a commonly agreed definition for user experience is still lacking (Kujala *et al.* 2011). The definition from ISO 9241-210 (2010) has been described as being too abstract and imprecise and that definition does not include definition of metrics (Law 2011). This is one of the current challenges for the field of HCI and for examining user centred evaluation. A commonly agreed definition is needed particularly to be able to compare results from many research studies. A common conceptual framework to base these studies on is needed. Furthermore it would be useful for IT professionals to have a definition of commonly used measurements for user experience (Law 2011).

In this thesis the definition from the ISO 9241-210 (2010) is used as a definition of user experience, including the three notes described, because that definition is relatively new and has been accepted in an international standard.
2.3 Remarks on the Concepts

In this thesis the concept *user centred evaluation* includes both evaluating usability and evaluating user experience. Some methods are used specifically to evaluate usability, called usability evaluation methods, and others are used to measure the user experience in particular, called user experience methods. In some evaluations both usability and user experience are evaluated with the same users. User centred evaluation is used in this thesis as a general concept for all these types of evaluations.

In papers I and II, usability evaluation methods are studied, where usability problems are used as a measurement for the usability of the software evaluated in the evaluations. In paper III, IT professionals in Iceland were asked about usability testing. It should be noted that in Icelandic the same word “nytsemisprófanir” is used for usability testing and usability evaluation. Therefore, the IT professionals involved define usability testing and usability evaluation as being the same activity. In paper IV, IT professionals were asked how they gather feedback on their design, the user requirements and the context of use, from the user’s perspective. In paper IV the view on evaluation has a wider scope than in the other papers, including both gathering feedback on the usability of the software and the user experience of using it.
3 Examining User Centred Evaluation

The research goal when examining user centred evaluation is often to study some factors of the evaluation. Two alternatives for researching the factors are conducting an experiment for examining the factors and observing some factors in more practical settings. In this chapter the context of the evaluation activity is described and a model for describing the factors of user centred evaluation is illustrated and explained. Two research approaches are described to examine user centred evaluation and finally an overview is given of what factors are studied further in this thesis.

3.1 The Context of User Centred Evaluation

User centred evaluation is conducted in the context of software development. It is one of many activities that IT professionals undertake and there are different stakeholders involved. In Figure 1 the context of evaluation is illustrated and the various factors in the figure are described briefly below.

![UCE model](image)

**Figure 1. The context of user centred evaluation.**
To illustrate a point of reference in this thesis, a model was developed called the UCE model, which describes the factors involved when conducting user centred evaluation. The UCE model will be described in subchapter 3.2.

According to Woolrych et al. (2011) there are a number of factors that relate to the client, to the project manager and to the usability practitioners that influence the configuration of the evaluation. The presentation in Figure 1 is based on these factors. Additionally, a description of the design context from Svanæs and Gulliksen (2008) was used as an inspiration for the illustration of the context of user centred evaluation. The factors affecting evaluation are explained briefly below.

**Client factors:** The client factors include the needs and expectations of the client, the design vision for the product and the design purpose, the criteria used for prioritising the product, the context of the business of the client, budgetary resources, such as time and money and other logistical resources. The representatives from the client communicate these issues to the project manager or to the usability professionals.

**Project factors:** The project factors include project leadership, the development process used in the project, the development stage, where the evaluation is conducted and prioritization criteria for the project. In addition, the competences and experiences of the project team members affect the software development. The main person responsible for taking care of these factors is the project manager. Software development will be described further in Chapter 5.

**Usability work factors:** The relation of usability work to the overall software development approach affect the user centred evaluation. Moreover, the experience and competence of the usability professionals affects the evaluation, and the maturity and motivation of the usability professionals. Defined usability criteria could also influence the evaluation activity. The context of use and the user requirements are two of the main outcomes of user centred design activities according to ISO 9241-210 (2010).

These three groups of factors influence the actual evaluation activity. The effect of using the Scrum software development process on user centred design activities is studied in papers III and IV.
3.2 The UCE Model

The factors that relate directly to the user centred evaluation activity are illustrated in a model called the UCE model in Figure 2, which illustrates a point of reference in this thesis. The UCE model describes the factors involved when conducting user centred evaluation, called the UCE factors. A short description of each UCE factor will be given below. The selection of the UCE factors is based on guidelines on how to run user centred evaluations from several publications (Wixon and Wilson 1997, Kwahk and Han 2002, Stone et al. 2005) and on the ISO/IEC 25026 standard (2006) and the ISO 9241-210 standard (2010).

![UCE model diagram]

Figure 2. The UCE model for illustrating user centred evaluation factors.
Description of the UCE factors:

1. **Purpose of the evaluation**: The purpose of the evaluation describes why the evaluation is conducted. The purpose could be to gather feedback on the usability of the software and the user experience as described in subchapter 2.2. Additionally, feedback on new user requirements and of the context of use could be gathered. The purpose of an evaluation conducted by IT professionals using the Scrum process is studied in paper IV.

2. **Evaluation plan**: The evaluation plan describes the procedure of the evaluation by listing the subtasks of the evaluation that need to be conducted to perform the evaluation. For each subtask it is stated what needs to be done, by whom and when it will be done.

3. **Evaluation method**: An evaluation method is a collection of particular subtasks, which should be conducted during the user centred evaluation activity to be able to say that a particular evaluation method has been used. Evaluation methods will be described in more detail in subchapter 4.1.1. Knowing what evaluation method will be used is important when planning the evaluation. The effectiveness of using evaluation methods is described in subchapter 4.1.2. The effectiveness of using three different methods is studied in paper I. The type of user centred evaluation conducted is studied in paper IV.

4. **Material for empirical evaluation**: User tasks that the participants are asked to perform during the evaluation are often used, when conducting an evaluation with user participation. In other cases, scenarios could be used and instructions on what to do in the evaluation could be given. Additional material presented to the participants could be a consent form and documents describing the goal of the evaluation.

5. **Material for analytical evaluation**: Guidelines and standards could be used while conducting some types of inspection evaluation, which are often identified by the material used, like heuristic evaluation and standards inspection. The use of two different sets of
guidelines, which were used as supporting material when conducting heuristic evaluation, was studied in paper II.

6. **Support on the method:** The subtasks that need to be achieved while using the method could be described in a written document to support the evaluators when using the method. In addition, training and tutorial support on the evaluation method could be used during the evaluation.

7. **Software:** The goal of software development is developing software, which is defined as a set of computer programs, procedures and associated documentation and data according to ISO 12207 (2008). During development the software can have various stages of design when evaluated, it can be of various types and some part of the software could be implemented and some not. These three attributes of the software are numbered 7a, 7b and 7c and will be described below. The status of the three attributes influences the evaluation and what methods can be used to evaluate the software.

7a. **Stage of design:** The software being evaluated could have various stages of design. It can be low fidelity prototype on paper of the first ideas of the software; it can be detailed designed in a tool or something there in between. The stage of the design of the software can constrain how the evaluation is conducted.

7b. **Type:** The type of the software describes what the main field for using the software is. Some examples are business software, a game, social networking software and information retrieval software. The type of the software can influence what evaluation methods can be used and the measurements that can be gathered during the evaluations.

7c. **Part of software:** It needs to be clear what parts of the software are being evaluated during the evaluation. Sometimes a prototype illustrates a small part of the software, sometimes the prototype is extensive and sometimes all the software is designed and could be evaluated.
8. **Equipment used for the evaluation:** The equipment used in the evaluation to illustrate the software can have many forms. One example is using a computer to run a detailed design; another is using foam to illustrate a computer or a phone.

9. **Evaluation environment:** An evaluation can take place in various surroundings. One possibility is conducting an empirical evaluation in the real settings, often called in the field, where the software is actually used. Evaluation can be conducted in more controlled environment like in a usability laboratory. Expert evaluation often takes place in the development environment.

10. **Users:** In empirical evaluation, real users of the software can take part in the evaluation or user surrogates are sometimes asked to participate. Sometimes there are real users participating, which need to talk to secondary users that also participate in the evaluation. In analytical evaluation, users do not participate, but the evaluator should have their characteristics in mind while evaluating. User participation in evaluations is studied in paper IV.

11. **Evaluator:** The evaluator plans the evaluation, conducts it and has a major role while analysing the data to describe the results from the evaluation. What types of user centred evaluation are conducted by the different roles of evaluators is studied in paper IV. The roles of the people conducting usability evaluations are studied in paper III. The effect of the evaluators’ knowledge on the results of the evaluation is described in subchapter 4.1.5.

12. **Possible observers:** Sometimes IT professionals and representatives from the client observe when an empirical evaluation is conducted. That experience can give valuable information for deciding how to react to the results of the evaluation.

13. **Data:** The data gathered in the user centred evaluation could be both quantitative and qualitative feedback on the usability and the user experience of using a particular design, see Chapter 2 for further details. Additionally, feedback on new user requirements and the context of use can be gathered. Usually there is some equipment used to gather the data.
14. **Data gathering equipment**: Data could be gathered in the evaluation by writing notes, video recording, voice recording and by logging software usage to name the most common ones. Using paper versus using a software tool was studied in paper II in this thesis.

15. **Results**: The results of the evaluation depend on the purpose of the evaluation and the method selected. The gathered data is analysed, interpreted and results are presented. The results from measuring usability by counting usability problems are studied in papers I and II.

16. **Recipients**: It depends on the evaluation purpose what the roles of the IT professionals receiving the results from the evaluation are. Sometimes it is the evaluator who conducted the evaluation that uses the results and decided actions and sometimes the results are handed over to a person, having another role, who decided how to react to the results of the evaluation.

17. **Decisions**: The recipients of the results from the evaluation need to decide what actions to take on the results. Sometimes it is decided to redesign according to the results, sometimes the help provided in the software is extended, and sometimes no action is taken.

All these UCE factors can vary while conducting user centred evaluation. Experiments can be conducted to gain knowledge on how much it affects the evaluation if there is a variance in one or a known number of the UCE factors of evaluation. Moreover, the UCE factors can be studied in practical settings, gaining knowledge on how IT professionals choose to conduct user centred evaluation according to the various UCE factors. These two research approaches will be described in subchapters 3.3 and 3.4.

### 3.3 Researching Evaluation in Experimental Settings

An experimental approach to research was used in papers I and II in this thesis. When researching user centred evaluation in experimental settings fundamental concepts are used, which will be defined below. The term *experiment* as defined by Oehlert (2000) is:
An experiment has treatments, experimental units, responses, and a method to assign treatments to units (p 4).

The concepts in this definition are treatments, experimental units, responses and a method. The method for assigning treatments to units in experiments is randomization. The concepts are defined in (Oehlert 2000):

Treatments are the different procedures we want to compare (p. 6).

Experimental units are the things to which we apply the treatments (p. 6).

Responses are outcomes that we observe after applying a treatment to an experimental unit. That is, the response is what we measure to judge what happened in the experiment; we often have more than one response (p. 6).

Randomization is the use of a known, understood probabilistic mechanism for the assignment of treatments to units. Other aspects of an experiment can also be randomized: for example, the order in which units are evaluated for their responses (p. 6).

When comparing evaluation methods in an experiment, like was done in paper I in this thesis, the treatments are the evaluation methods used in the study, the experimental units are the evaluators using a particular evaluation method and the responses are measured by counting usability problems found when using a particular method. Randomization was used to randomly assign the evaluation methods to the evaluators, so no systematic error occurs based on the assignment of the evaluation methods.

Another example would be to manipulate the supporting material used during the evaluation, so two groups of evaluators could be asked to use two different sets of guidelines while using a method called heuristic evaluation, like was done in paper II. Additionally, the tool for registering the data gathered in the evaluation could be manipulated asking half of the participants to write on paper and the other half to use a software tool to register the outcome. This was also done in paper II.

One of the fundamental concepts in experimental research is control (Oehlert 2000):

Control has several different uses in design. First, an experiment is controlled because we as experimenters assign treatments to experimental units. Otherwise, we would have an observational study. Second, a control treatment is a “standard” treatment that is used as a baseline or basis of comparison for the other treatments (p. 7).

The treatments used in paper I and II were controlled carefully in the studies. Moreover, in paper I, control groups were asked to evaluate
软件，其中其他组没有评估，因此这些评估者可以作为一个基准进行比较。

控制是将实验与从观察当前行为的实验中区分开的主要因素。观察性研究常用于在实际环境中研究评估。

3.4 研究评估在实际环境中的研究

当在实际环境中研究用户中心化评估时，研究人员观察当前行为。研究人员不操纵评估的任何因素，而是观察评估的因素。这种做法在论文III和IV中使用，使用问卷调查和半结构化访谈作为数据收集方法来观察IT专业人士的当前行为。

观察性研究的关注点可能是观察评估是如何进行的。Nørgaard和Hornbæk（2006）研究了14次当面实验，由研究人员观察并分析结果。该研究提供了关于使用当面实验方法的实用问题以及如何在实际环境中使用该方法的洞察。

IT专业人士可以被问及他们在实际环境中使用哪些方法；他们在观察用户中心化评估时所扮演的角色，以及评估的目的是什么，就像在论文IV中所做的那样。此外，可以分析评估是在什么背景下进行的，例如IT专业人士所从事的其他活动，他们正在使用什么样的软件开发流程，他们工作的公司有什么特点，以及他们专业背景的特点。

研究评估在实际环境中的研究也可以在研究中进行，研究者不仅是一个观察者，也是一个积极的参与者，就像在民族研究中所做的。研究者的参与范围从完全参与者到完全观察者（Gold 1958）。作为完全参与者的研究者可以亲身经历研究人员的反应，但风险是她可能会失去自己的视角并变得有偏见。另一方面，如果研究者是一个完全的观察者，她

23
remains detached from the subjects and does not influence the study as much, but there is a greater risk of misinterpretations (Gold 1958).

The main difference of doing experimental and observational research on user centred evaluation is that when using the experimental approach the researcher manipulates a small number of factors, while in the observational one, the researcher does not manipulate any factor, but observes particular factors and analyses the relation between them.

3.5 Overview of the UCE Factors Studied

An overview of the UCE factors studied in this thesis is given in Table 1. The table describes what research approach is used to study the UCE factor, in what chapter the study is described and in what paper further details can be found.

<table>
<thead>
<tr>
<th>UCE factor</th>
<th>Research approach</th>
<th>Explained in chapter</th>
<th>Studied in paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client factors</td>
<td>Practical settings</td>
<td>5</td>
<td>III and IV</td>
</tr>
<tr>
<td>Usability work factors</td>
<td>Practical settings</td>
<td>5</td>
<td>III</td>
</tr>
<tr>
<td>Purpose of evaluation</td>
<td>Practical settings</td>
<td>5</td>
<td>IV</td>
</tr>
<tr>
<td>Evaluation method</td>
<td>Experimental and practical settings</td>
<td>4, 5</td>
<td>I and IV</td>
</tr>
<tr>
<td>Material for analytical evaluation</td>
<td>Experimental settings</td>
<td>4</td>
<td>II</td>
</tr>
<tr>
<td>Data gathering equipment</td>
<td>Experimental settings</td>
<td>4</td>
<td>II</td>
</tr>
<tr>
<td>Evaluator</td>
<td>Practical settings</td>
<td>5</td>
<td>III and IV</td>
</tr>
<tr>
<td>Users</td>
<td>Practical settings</td>
<td>5</td>
<td>IV</td>
</tr>
<tr>
<td>Results</td>
<td>Experimental settings</td>
<td>4</td>
<td>I</td>
</tr>
</tbody>
</table>

In Chapter 4, user centred evaluation is examined in experimental settings. The studies are described in papers I and II. The related work is described and the experimental studies are summarised and discussed. The UCE factors studied are the evaluation methods used, the material for analytical evaluation, the data gathering equipment and the evaluation results.

In Chapter 5 user centred evaluation is studied in practical settings. The studies are described in papers III and IV. A structure is used that is similar to the one employed in Chapter 4. The UCE factors studied are: the client factors, the usability work factors, the purpose of the evaluation, the evaluation methods, the evaluator and the users.
4 Examining Evaluation in Experimental Settings

The content in this chapter relates to the first research question:

How different are the results of using various user centred evaluation methods for evaluating the same software?

Two studies were conducted to examine the first research question, which are described in papers I and II. In this chapter the related work that connects to the research question is described. Secondly the data gathering in the two studies described in papers I and II is described. Third, the results from the studies are summarised, compared to the related work and discussed. In Table 2 an overview of the studied UCE factors in papers I and II is given.

<table>
<thead>
<tr>
<th>UCE factor</th>
<th>Fundamental question</th>
<th>Studied in paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation methods</td>
<td>How effective are the various methods?</td>
<td>I</td>
</tr>
<tr>
<td>Material for analytical evaluation</td>
<td>Does the material used affect the results?</td>
<td>II</td>
</tr>
<tr>
<td>Data collection</td>
<td>Does the data collection affect the results of the evaluation?</td>
<td>II</td>
</tr>
<tr>
<td>Evaluation results</td>
<td>How different are the results from using three methods?</td>
<td>I</td>
</tr>
</tbody>
</table>

In paper I the quantity of usability problems found by using three evaluation methods was studied. Additionally, the quantity of serious problems found by using one of the three methods was compared. The effect of using two sets of usability guidelines when conducting heuristic evaluation and the effect of using two ways of collecting data on the usability problems found in the evaluations were described in paper II.

4.1 Related Work

Several research studies have been conducted, in which a particular UCE factor of evaluation has been studied and how changes in that factor affect the outcome of the study. Some of those studies are described in this subchapter. Related work is described in this subchapter on the four UCE factors studied in this thesis in experimental settings, which are: Evaluation
methods, material used in heuristic evaluation, data gathering equipment and evaluation results. In addition, related work on the effect of the evaluator knowledge when conducting an evaluation is described, because that is particularly relevant when discussing the results in this chapter.

4.1.1 User Centred Evaluation Methods

For the last 20 years many methods for evaluating software have evolved. A method description gives the expectation that it can be used by IT professionals in prescribed series of stages, with defined decision points and guidance on what action to take at each point (Woolrych et al. 2011). Some HCI specialists would like to use the word technique as opposed to the word method when talking about the evaluation activity, because many of the evaluation methods are not very structured (Stone et al. 2005). A method is actually more like a collection of resources, and its usage must be adjusted to the specific context that the IT professionals are working in (Woolrych et al. 2011). In this thesis, the word method is used, because it is commonly used by researchers in the field (Woolrych et al. 2011).

Some authors categorize the methods according to whether users participate in the evaluation, so called empirical methods, and methods that do not involve users, so called analytical methods (Whitefield et al. 1991, Barkhuus and Rode 2007). This is also done in this thesis.

Empirical Evaluation Methods

There are mainly three categories of evaluation methods where users participate in the evaluation:

(a) methods for measuring users’ performance,

(b) methods for observing users,

(c) methods for asking users for their opinions.

The empirical evaluations conducted in paper IV are analysed according to these groupings.

When evaluating users’ performance, the users are asked to perform some predefined tasks in controlled settings and the performance is measured. Typically, the time it takes to solve each task is measured, the number of errors made is logged and the navigation path through the interface is tracked. This type of evaluation is often called user testing.
Methods for observing users do always include an observer that observes how a user interacts with the software. The main methods for asking users for their opinions are interviews and questionnaires.

Actually all three categories of evaluation through user participation could be used with the same participant in user centred evaluation, that is:

(a) *Measuring the performance* - the performance could be measured during task solving.

(b) *Observing the user* - the user is observed during the task solving.

(c) *Getting users’ opinion* - the user could be interviewed for getting background information, and after the task solving the user could be asked to answer a questionnaire on his or her satisfaction with the software.

One example of conducting empirical evaluation is by using the think-aloud method, that was used in papers I and II.

**The Empirical Method Think-Aloud**

The think-aloud method was first introduced in software development around 1980 (Lewis and Mack 1982). There are numerous variations of the method that have been employed. The variant of the method that was used in paper I is described by Molich (2000) and Nielsen (1993). In that variant one user at a time is asked to solve predefined tasks and talk while he is interacting with the software, so the evaluator can understand how the user thinks about the software while using it. The evaluator conducts the evaluation by handing out the relevant material. In the task solving session, she hands out the tasks one at a time. If the user is not talking the evaluator encourages the user to say what she is thinking.

There are mainly five steps in this variant of the think-aloud method:

(a) *Greeting* the participant,

(b) *Data gathering* on the participant’s background,

(c) *Task solving session*, where the participant interacts with the software solving predefined tasks,

(d) *Debriefing* from the participant,

(e) *Thanking* the participant for coming.
A typical usability evaluation using this variant of the think-aloud method consists of evaluation sessions with five to ten participants. Each evaluation session lasts typically for one to two hours.

**Analytical Evaluation Methods**

It can be tedious to carry out empirical evaluations. Consequently, a number of analytical methods have been proposed over the last 20 years for evaluation through inspection where users are not involved in the evaluation. The most commonly described inspection methods in HCI textbooks are heuristic evaluation and cognitive walkthrough (Dix *et al.* 2003, Stone *et al.* 2005, Rogers *et al.* 2011). Moreover, using models to predict user performance is an expert-based approach to evaluation. These techniques are successful for software with limited functionality such as telephone systems. The keystroke level model and GOMS are best known in this category (Card *et al.* 1983).

In evaluation through inspection, the evaluator inspects the interface and assesses the impact it would have on the particular users. The evaluators use interface guidelines, user interface standards, the users’ tasks and their own knowledge, as a supporting material to base their evaluation on depending on the method. The results of using a particular inspection method are often described by a list of possible usability problems that the users would have, if they were interacting with the software, see subchapter 2.2.1.

**Heuristic Evaluation**

The heuristic evaluation method was used in papers I and II. The procedure of heuristic evaluation involves having a small group of evaluators examining the software individually using a predefined list of heuristics as supporting material in the evaluation. Typically the list of heuristics contains ten heuristics. In papers I and II a set of heuristics suggested by Nielsen (1993) was used. Additionally, a set of heuristics from Gerhart-Powals (1996) was used in paper II.

After evaluating individually the group of evaluators meets and aggregates the results to one aggregated list of usability problems. The heuristic evaluation was first introduced by Nielsen and Molich (1990),

**Cognitive Walkthrough**

The analytical method cognitive walkthrough was used in paper I. Cognitive walkthrough is an evaluation method that focuses on the user’s cognitive activities, goals and knowledge, when she is learning to solve particular tasks by using the software. During the walkthrough the evaluator steps through the actions in the user interface which are needed to solve a task and evaluates the possible usability problems that could occur. Usually the cognitive walkthrough is done in pairs of two evaluators who tell a believable story about why this particular step in the interface is easy or hard to use for the user. The method was first described in Lewis et al. (1990) and evolved through some versions (Polson et al. 1992, Wharton et al. 1994) that were more or less structured and tedious to apply.

**Other Classifications of Evaluation Methods**

Evaluation methods can also be classified according to several other orthogonal dimensions, including formal and controlled as opposed to informal and opportunistic (Twidalen et al. 1994). An informal evaluation activity indicates that the evaluation is conducted without fixed rules and does not need to produce an official result. On the contrary a formal evaluation is more structured, having predefined subtasks.

The results can be used to give IT professionals feedback on their current version of the user interface design which is still under construction, called formative evaluation. Evaluation results can also be used to assess the success of a finished product by gathering quantitative data, called summative evaluation (ISO 1998).

Barkhuus and Rode examined what evaluation methods had been used in research projects over a period of more than two decades (Barkhuus and Rode 2007). They group evaluation methods according to whether the results from the evaluation are quantitative or qualitative, sometimes also called objective or subjective (Hornbæk 2006). The second dimension from Barkhuus and Rode (2007) is whether the evaluation is empirical or analytical. An empirical qualitative evaluation could be interviewing users and gathering feedback during meetings. Qualitative
analytical methods could be peer reviewing the software and quantitative analytical method could be gathering usability problems with heuristic evaluation. The results show that empirical evaluation is conducted in over 90% of the research studies performed in 2006. In the remaining 10% of the studies, analytical evaluation, informal evaluations or no evaluation at all is conducted. Over 70% of the empirical studies are quantitative, whereas 12% of those also include qualitative measures and the remaining 30% are purely qualitative. The use of evaluation methods by IT professionals in practical settings has also been examined in various studies; see subchapter 5.2.1.

4.1.2 The Effectiveness of Using Evaluation Methods

Shortly after the first inspection methods were defined, researchers were interested in measuring the effectiveness of using various methods and finding the advantages and disadvantages of using inspection methods compared to using empirical methods. The most common way of estimating the effectiveness of the methods was by counting usability problems found and the severity of the problems found was studied.

Four studies comparing evaluation methods were published in the years 1990 to 1993, (Jeffries et al. 1991, Karat et al. 1992, Desurvire et.al. 1992, Nielsen and Phillips 1993). Three of these studies are better described and discussed in paper I. In addition, a study by Cuomo and Bowen (1994) is also discussed there. In these studies an aggregated list of all problems found during user observation is made and used to describe all usability problems that can be found in the software. Then the number of problems found by using another method is compared to the aggregated list. It is common to presume that problems found while observing users in user centred evaluation are true problems that users would have in real use. Problems found by using another method are compared with the list of problems found in the user observation to calculate the effectiveness of the method.

Gray and Salzman (1998) published a provocative paper, where the authors found methodological flaws in all the five studies published in 1990 to 1993. The authors recommend that researchers in the field pay close attention to experimental design in their studies on evaluation methods.
Studies where evaluation methods were evaluated were disregarded in the literature study of evaluation methods used in research studies by Barkhuus and Rode (2007), because these were meta-analysis papers and not seen as relevant in their study. The meta-analysis papers were 14% of the papers published in 1983; 3% of papers in published in 2000 and no paper had that focus in 2006. This trend is maybe a response to the critique from Gray and Salzman (1998), that the outcomes of the studies where evaluation methods are compared are not reliable and valid.

4.1.3 Supporting Material in Heuristic Evaluation

The effectiveness of using particular set of heuristics in heuristic evaluation can be studied by counting usability problems found and by interviewing the evaluators to get their opinion on the effectiveness. This was for example done in a study examining if heuristics used in heuristic evaluation are applicable for evaluating three Web 2.0 applications (Thompson and Kemp 2009). In that paper, results are presented from conducting heuristic evaluation using the traditional heuristics from Nielsen (1993) as supporting material in the evaluation. Moreover, two guidelines were added to the set of heuristics used in the evaluation to fit the Web 2.0 context. The results using the heuristic evaluation were compared to the results of empirical evaluation. The results from the heuristic evaluation showed that each of the three applications evaluated failed in some way. On the contrary the results from the empirical evaluation showed that user found the sites logical and easy to use. Based on the outcome of the research study, the authors conclude that the set of heuristics used for heuristic evaluation for Web 2.0 need to be adjusted to that context and suggest new heuristics to be used (Thompson and Kemp 2009).

Two playability heuristic sets were compared in a research study to discover their strengths and weaknesses when they are used to evaluate a mobile game using the heuristic evaluation method (Korhonen et al. 2009). Data was gathered both through group interviews with the evaluators and by analysing the reports from the evaluations. The results indicate differences in clarity and comprehensibility. The authors conclude that both heuristic sets should be improved to provide more precise and
relevant evaluation results when evaluating videogames with the heuristic evaluation method.

The results from these studies show that the chosen set of heuristics used during heuristic evaluation can affect the results of the evaluation and evaluators need to consider what heuristics are appropriate for the context of their evaluation.

### 4.1.4 Data Gathering Equipment while Conducting Evaluation

The data collected in user centred evaluation can be registered using a paper or using a digital tool. Digital documents can easily be searched, shared, stored, accessed remotely and linked to other related material, whereas paper has a unique set of affordances such as being tangible, and spatially flexible (Gladwell 2002).

Some digital tools have been suggested to gather and analyse the data from user centred evaluations. One example is a tool suggested to assist evaluators in defining the tasks, specifying the heuristics, and documenting usability problems when conducting heuristic evaluation described by Pribeanu (2009). The tool proved useful as a training tool for novice evaluators because of the possibility to provide more assistance during the evaluation activity than using a paper. Moreover, the results show that the tool provided better understanding of usability activities.

Some tools are designed for particular types of user centred evaluations. A visualization tool was made for pre-processing and visual analysis of data from a usability study of mobile indoor navigation application for visually impaired users (Maly et al. 2010). Their results show that the participants in the study were able to find similar amount of usability problems by using the tool as without using the tool, but the tool increased the pre-processing speed. Moreover, the time necessary for analysing the results was reduced.

### 4.1.5 Evaluator’s Knowledge of Evaluation Methods

Some studies have shown that, if many evaluators evaluate the same software, they do not necessarily find the same usability problems. Hertzum and Jacobsen (2001) performed a review of eleven studies of the evaluator effect while using three methods: cognitive walkthrough, heuristic evaluation and the think-aloud method. Their main conclusion is
that the average agreement between any two evaluators, who have evaluated the same software with the same method, ranges from 5% to 65%. None of the three methods involved in the study was constantly having higher agreement than the others. The main reasons for this were vague evaluation goal analyses and vague evaluations procedures. This meant that the evaluators used the methods in various ways and employed vague criteria for what should be stated as a usability problem.

In a study on heuristic evaluation, Nielsen (1992) examined how much the evaluator’s knowledge of usability in general and the evaluator’s knowledge of the kind of interface being evaluated affected the evaluation results. Usability specialists found more usability problems than novice evaluators in that study and the double experts that had extensive knowledge of usability and the kind of interface being evaluated found the highest quantity. The effect of using two cognitive approaches, field dependent and field independent, on the results from heuristic evaluation was examined in the study (Ling and Salvendy 2009). The results show that the independent cognitive style, where the evaluators are analytically-minded and take active approaches while solving problems, had significantly higher thoroughness, validity, effectiveness, and sensitivity than the field dependent cognitive style, where the evaluators are less analytically-minded. The analytically-minded evaluators also felt it was easier to find problems than the field dependent evaluators.

The consistency of usability testing across nine industrial organizations was studied (Molich et al. 2004). The evaluators worked in nine teams and found 310 different usability problems in total. Only two problems were reported by six or more teams, while 75% of the problems were uniquely reported, meaning that were only reported by one team. The authors conclude that the assumption that if evaluators use the same method they will obtain the same results is plainly wrong.

Information on the evaluators’ knowledge of usability in general and their knowledge of the type of interface evaluated should be gathered when conducting and researching user centred evaluation. Additionally, data should be gathered on other UCE factors like the method used during the evaluation, the supporting material used during the evaluation, the data collection equipment used and the definition of the results of the evaluation, because all these UCE factors could influence the results of the
evaluation. Information on the activity of analysing the data and reporting the results is also important.

4.2 Data Gathering and Analysis

In this chapter the data gathering conducted in papers I and II and the analysis approach are described. In both these papers data were gathered on the usability problems found. The data were analysed and a joint list of problems was made.

4.2.1 Gathering Usability Problem Lists

In papers I and II an experimental design was used to gather usability problem lists with the purpose of comparing the results from evaluations using three different usability methods.

In paper I, 51 computer science students used first either heuristic evaluation according to Nielsen’s guidelines (Nielsen 1993) or cognitive walkthrough for one week in groups of three evaluators. Two types of software were evaluated; experimental text retrieval software called TeSS was evaluated by 33 students and a graphical text editor Asedit was evaluated by 18 students. After that each group handed in a usability problem list. Then all the groups evaluated the software TeSS for one week using the think-aloud method with three users, who were also computer science students. The students made the tasks for the think-aloud evaluation themselves and they chose the evaluation environment. The evaluators had not used the methods before but obtained detailed material describing the methods. They could all be grouped as novice evaluators. This experiment was part of a grading in a HCI course that the students were having on their third year of studies. Therefore they were quite motivated to carry out a thorough evaluation. The recipients of the results were the authors of the paper and one of those was their lecturer. The experimental design in this study included various opportunities for comparing the results both from using one method and combining the results from using one method after the other.

In paper II results from using two types of guidelines in heuristic evaluation, Nielsen’s guidelines (Nielsen 1993) and Gerhart-Powals’ guidelines (Gerhardt-Powals 1996) are described. In addition, results from reporting usability problems using paper and with the help of a web tool
are compared. The 20 evaluators using the heuristic evaluation were computer science students who had not used the method before. They were asked to participate in a research project, but did not obtain any grading or reward after taking part in the study.

The think-aloud evaluations were conducted by two students who had more knowledge of the evaluation method than the evaluators using the heuristic evaluation. They ran ten think-aloud evaluations with predefined tasks that were defined by the researchers specially to be able to find the problems found in the heuristic evaluations. The think-aloud evaluations were conducted with real users in their real work environment. The think-aloud evaluations were graded as a part of a course the students were taking, so the students were highly motivated to perform the study. The recipients of the results of the evaluations using the heuristic evaluation and the think-aloud evaluations were the authors of the paper.

4.2.2 Analysing the Effectiveness of the Evaluations

All the documents describing the usability problems found were collected in the studies described in papers I and II. The goal was to estimate the effectiveness of the evaluations by analysing how many problems were found by using each method. The lists were analysed and one joint list was made. The activity of making the joint list was not trivial. First, all the problems found were gathered in one list. Then, for each of the problems, it was decided if it was the same as some other problem on the list, or if it was a unique problem. In paper I the approach for joining the lists was decided by the authors of the paper prior to the analysis and applied systematically. In paper II a systematic and pre-described approach to make the joint list was taken as described in Connell and Hammond (1999). The researchers made their own judgment of which two problems were the same in both papers. Additionally, the severity of each problem was judged by the researchers. When comparing results from one study to those from another, the approaches used in making the joint list and judging the severity have to be similar to make a fair comparison of the effect of using one method or the other.

Gray and Salzman, (1998) suggest that usability problems found in an evaluation are categorized in four categories, which are used in paper II.
The suggested categories are:

(a) *Hit*: Problem is found and it is a true problem.

(b) *False Alarm*: Problem is found but no problem exist.

(c) *Miss*: A problem is not found but it exist.

(d) *Correct rejection*: A problem is not found and does not exist.

It is common to presume that problems found while observing users in user centred evaluation are true problems that users would have in real use. Problems found by using analytical methods can either be hits, which are problems also found in empirical evaluations, or false alarms, which are problems not found in empirical evaluations. To measure the effectiveness of using a particular analytical method, the problems found using that method are compared with the list of problems found in user observations.

Hartson et al. (2001) describe measures for studying evaluation methods as:

**Reliability**: Evaluators want consistent usability evaluation results, independent of the individual performing the usability evaluation (p. 14).

**Thoroughness**: Evaluators want results to be complete; they want evaluation methods to find as many of the existing usability problems as possible (p. 14).

**Validity**: Evaluators want results to be “correct”; they want evaluation methods to find only problems that are real. (p. 15).

Hartson et al. (2001) defined formulas for calculating the thoroughness and the validity to be able to estimate these factors when conducting research on evaluation methods. When the definition from Gray and Salzman (1998) on hits, misses and false alarms are combined with the definition from Hartson et al. (2001) on thoroughness and validity the definition are:

\[
\text{Thoroughness} = \text{Hits} / (\text{Hits} + \text{Misses})
\]

\[
\text{Validity} = \text{Hits} / (\text{Hits} + \text{False alarms})
\]

Hartson and colleagues (2001) also added a new one, called effectiveness:

\[
\text{Effectiveness}^5 = \text{Validity} \times \text{Thoroughness} \quad (p. 19)
\]

---

5 The effectiveness defined here is not the same as the effectiveness factor defined for usability in subchapter 2.1.
The evaluators in the study described in paper I were asked to hand in documents describing how much time it took to use the particular evaluation method to be able to calculate the efficiency in using the methods. The evaluators handed in the time it took for each group to prepare, evaluate, analyse and describe the results for each method. These data were analysed by calculating the average time for each group and then the statistical difference was calculated with an F-test.

4.3 Results

In this subchapter the goal of the studies described in papers I and II are described and the results are summarised.

4.3.1 Paper I: Prediction of Usability

This paper is a presentation of an experiment where the effectiveness of three evaluation methods in uncovering and assessing usability problems was compared. The evaluation methods that were considered were cognitive walkthrough, heuristic evaluation and the think-aloud method.

The reason for comparing these methods, even though these could be used for different purposes in software development, is to understand how different the results are and what the strengths and weaknesses of using these methods are.

The goal was to study:

(a) If the results of using either heuristic evaluation or cognitive walkthrough were different from the results of using the think-aloud method for evaluating the same software.

(b) If the results from heuristic evaluation were different from the results of the cognitive walkthrough.

(c) If combining the results from using two evaluation methods would give higher quantity of usability problems than using only one method.

The quantity of usability problems found by using either heuristic evaluation or the think-aloud method was significantly higher than using the cognitive walkthrough method. Three evaluators using heuristic evaluation or conducting think-aloud evaluations with three users
uncovered on average 80% more usability problems than a group of three evaluators using the cognitive walkthrough method.

Heuristic evaluation and the think-aloud method were found to be complementary. If heuristic evaluation was used prior to using the think-aloud method, a higher number of usability problems were found during the think-aloud evaluations than the control groups, which had not used heuristic evaluation, found during their think-aloud evaluations. A group of inexperienced evaluators were capable of finding 61% of highly critical problems in average using the think-aloud method and 87% of highly critical problems in average using the heuristic evaluation. If heuristic evaluation was used prior to the think-aloud evaluations, the coverage was up to 80% in average when finding highly critical problems using the think-aloud method. Up to 25% of the severe problems were found using one method.

The main contribution in this paper is the results on the effect of using one usability method after the other. The results show that the use of heuristic evaluation before the think-aloud method improves the results of the think aloud evaluation substantially, especially when finding highly critical problems.

4.3.2 Paper II: Heuristic Evaluation

The aim of this paper was to refine a research agenda for comparing and contrasting evaluation methods, heuristic evaluation and the think-aloud method. To reach this goal, a framework was presented to evaluate the effectiveness of different types of support for structured usability problem reporting. This paper reports on an empirical study of this framework that compares two sets of heuristics, Nielsen’s heuristics (Nielsen 1993) and the cognitive principles of Gerhardt-Powals (1996), and two media of reporting a usability problem, i.e. either using a web based tool or paper.

In paper II the goal was twofold:

(a) To compare the number and seriousness of problems found per evaluator in heuristic evaluation with two different sets of usability heuristics, from Nielsen and Gerhardt-Powals.

(b) To compare two different ways of reporting usability problems, on paper and with the help of a web tool.
The validity of using Nielsen’s heuristics and Gerhardt-Powals heuristics was almost the same, around 0.40. This means that if the results of the think aloud evaluations are the baseline, around 60% of the effort of doing heuristic evaluation has been wasted. Similarly the thoroughness of the two sets of heuristics was 0.36. That means that 64% of the usability problems discovered in think-aloud evaluations were undetected by the evaluators in the heuristic evaluations. Percentage wise, when using the Gerhardt-Powals heuristics more severe problems were found, but the difference was not significant. The results also show that the validity was almost the same when using a paper and using a tool to report the problems, even though a higher quantity of problems was reported using a web based tool than using paper.

The main contribution of the paper is the description of a framework to evaluate the effectiveness of different types of support for structured usability problem reporting that can be reused by other researchers. Two case studies were analysed successfully according to the framework in the paper.

4.4 Discussion

In this subchapter the results in papers I and II are discussed and compared to related work.

The quantity of problems found by using the heuristic evaluation compared with using the think-aloud method from papers I and II does not match. The reason could be that the evaluators did not have the same skill level in these two studies, often referred to as the evaluator effect (Hertzum and Jacobsen 2001). The evaluators using the think-aloud method in paper II were more skilled than the evaluators doing the heuristic evaluation, but in paper I the evaluators using the three methods had similar background. Moreover in paper II, ten users were asked to participate in the evaluation using the think-aloud method, but five evaluators used the heuristic evaluation. The results from using the different methods in paper II are therefore not totally comparable. Further, the results from paper I are more reliable when comparing the quantity of usability problems found by each method.

In both these studies, the assumption was made that finding a list of all the real problems in the software is possible, if enough think-aloud
evaluations are conducted. That approach was used in paper II. In paper I there was a slightly different approach taken. The list of all usability problems in the software was made on the basis of the results from the think-aloud evaluations, but the authors added some problems to the list that were found by using the other methods and that the authors found important. When this list of all real usability problems has been made, calculations can be made on how many real problems the evaluation through inspection returns and how many false alarms and misses are reported. Chattratichart and Lindgaard (2009) state that it is unlikely that a usability evaluation will reveal all problems that exist. If that is true, a researcher that observes hundreds of think-aloud sessions can’t be sure that she has the list of all real problems in the software and then counting how many of those problems are found by other methods is not as reliable as it seems. Further research is needed on how reliable usability methods really are. What is significant is probably only the relative effectiveness of the evaluation methods within each study.

Wixon (2003) claims that the literature where usability evaluation methods are compared fails the practitioner. Moreover, he states that problem detection is only the first step in improving the software. It is not sufficient to make a list of the usability problems found. To improve the software the improvements need to be thought through using the usability problem list as a basis. He states that even though a method is very effective in finding problems, these problems would maybe not lead to changes in the software and therefore the evaluation method is not effective.

The quantity of problems found by using different evaluation methods were compared in the two studies described in papers I and II in this thesis. These studies were focusing on that outcome of the evaluations, which is finding usability problems. Other studies have covered how these outcomes can be described in an efficient way to help the IT professionals to decide what to do about the problems (Thorgeirsson and Larusdottir, 2007). The most efficient way of describing the results of the evaluation to the IT professionals was, for example, studied by Hornbæk and Frøkjær (2005). Their results show that IT professionals assessed redesign proposals to have higher utility than usability problem lists.
One way of determining the effect of using usability evaluation methods is to look at the downstream utility, which is defined by Law (2006) as:

*The extent the improved or deteriorated usability of a system can directly be attributed to fixes that are induced by the results of usability evaluations performed on the system.*

Here the effect of the usability evaluation is determined by how much it improves the actual usability of the software and not by how many problems are found. Researchers do not agree on the scope of usability evaluation. Cockton (2005) argues that assessing the downstream utility is beyond the scope of pure evaluation methods.

### 4.5 Lessons Learned

In papers I and II the evaluation methods were studied in experimental settings asking several evaluators to evaluate the same software and hand in lists of usability problems. The evaluation context was not completely realistic. The evaluators were computer science students that had neither much experience in preparing and conducting the evaluations nor in describing the results from the evaluations. The users were also computer science students in the study described in paper I and there were no constrains on the evaluation environment in that study. All this variation makes it hard to generalize how the results would be in realistic settings, where more experienced evaluators use the methods in their own software development.

The first research question in this thesis is: How different are the results of using various user centred evaluation methods for evaluating the same software? Taking into consideration the above assumptions, the results described in this chapter indicate that the answer to this question is that using the methods cognitive walkthrough, heuristic evaluation and the think-aloud method to evaluate the same software gives different results. Moreover, the results indicate that the use of the think-aloud method can be recommended to IT professionals because it has proven to be effective. Combining the use of heuristic evaluation and think-aloud is preferable because that is even more effective, especially when finding critical problems.
5 Examining Evaluation in Practical Settings

The content in this chapter relates to the second research question:

How do IT professionals conduct user centred evaluation in the Scrum development process?

Two studies were conducted to examine the second research question and are presented in papers III and IV. In this chapter background material on software development is described, together with related work on the UCE factors studied in papers III and IV. Secondly the data gathering in the two studies detailed in papers III and IV is discussed. Third, the results from the studies are summarised and compared to related work. In Table 3 an overview of the UCE factors studied in papers III and IV is given.

<table>
<thead>
<tr>
<th>UCE factor</th>
<th>Fundamental question</th>
<th>Studied in paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation methods</td>
<td>What evaluation methods are used in practical settings?</td>
<td>IV</td>
</tr>
<tr>
<td>Client factors</td>
<td>What are the dominant constrains, when conducting user centred evaluation?</td>
<td>III and IV</td>
</tr>
<tr>
<td>Usability work factors</td>
<td>How often is usability evaluation conducted?</td>
<td>III</td>
</tr>
<tr>
<td>Purpose of the evaluation</td>
<td>Why is the evaluation conducted?</td>
<td>IV</td>
</tr>
<tr>
<td>Evaluator</td>
<td>What types of user centred evaluation are used by each evaluator’s role?</td>
<td>III and IV</td>
</tr>
<tr>
<td>Users</td>
<td>Are the participants in the evaluation real users or user surrogates?</td>
<td>IV</td>
</tr>
</tbody>
</table>

The evaluation methods used in practical settings are analysed in paper IV. In addition, the types of user centred evaluation, which are conducted by each evaluator’s role, are studied in both in paper III and paper IV. The client factors dominant for constraining evaluations are investigated in paper III and paper IV. The frequency of conducting evaluations is studied in paper III. The purpose of the evaluation and the type of participants in empirical evaluation are considered in paper IV.
5.1 Background on Software Development

In this thesis the user centred evaluation activity is examined particularly in papers III and IV when conducted in the context of the Scrum development process. In this subchapter background material on software development processes will be given, then the agile software development process Scrum will be described together with the human-centred design approach.

5.1.1 Software Development Processes

Definitions of software development process terms will be given in this subchapter. Additionally, different types of software processes and related work on software processes are described.

Definition of Software Development Process Terms

It is stated in the introduction to the ISO 12207 (2008) that:

This International Standard establishes a common framework for software life cycle processes, with well-defined terminology, that can be referenced by the software industry (p. 1).

Software life cycle is explained in the standard by the following text:

The life cycle begins with an idea or a need that can be satisfied wholly or partly by software and ends with the retirement of the software. The architecture is built with a set of processes and interrelationships among these processes (p. 11).

A process and related terms are defined as:

Process: Set of interrelated or interacting activities which transforms inputs into outputs (p. 5).

Activity: Set of cohesive tasks of a process (p. 3).

Task: Requirement, recommendation, or permissible action, intended to contribute to the achievement of one or more outcomes of a process (p. 8).

Process outcome: Observable result of the successful achievement of the process purpose (p. 5).

Hence, a software life-cycle process is according to the ISO 12207 (2008) a set of activities, each of which is a set of tasks of a process with observable results and successful achievements.
In the standard a *life cycle model* is also defined (ISO 2008):

**Life cycle model**: Framework of processes and activities concerned with the life cycle that may be organized into stages, which also acts as a common reference of communication and understanding (p. 4)

It is stated in the ISO 12207 (2008) standard that:

*Application of a life cycle model provides the means to establish the time-dependent sequence necessary for project management* (p. 12).

Hence, a life cycle model is used to divide the software development projects into stages that are performed in a particular order. The model is also used by the IT professionals so they understand the management of the software project and can communicate about the status of each activity.

A life cycle model is often called a software development process (Boehm 2006) and this is also done in this thesis. Among the information that IT professionals ordinarily want to extract from descriptions of software development processes are information on: what is going to be done, when and where it will be done, how and why it will be done, who is going to do it, and who is dependent on it being done (Curtis *et al.* 1992).

**Types of Software Development Processes**

Various types of software development processes have been described. The main types are sequential software development and iterative software development.

The sequential software process is often named the Waterfall process (Boehm 2006). One of the first versions of the Waterfall model was described by Royce (1970). In its simplest version there are four stages in the process: requirements analysis, design, coding and testing. The fundamental thought in this process is that it is sequential, so that each stage should be more or less finished before the next begins. Seven stages are described in the original description of the development process by Royce (1970): two requirement stages, followed by analysis, system design, coding, testing and operational stage. It is suggested though, that succeeding and preceding stages are iterated if some difficulties occur. The Waterfall model has most frequently been interpreted as being purely sequential, partly due to convenience in contracting for software development (Boehm 2006).
In the 1990’s more emphasis in software development was on delivering parts of the software to customers iteratively and incrementally than before, so time to market would be shorter (Boehm 2006). Additionally, the software was getting more interactive with emergent requirements. In incremental software development the software requirements are divided into parts, which are implemented and a deliverable version of that part of the software is made (Basili and Turner 1975). The basic idea is that the IT professionals use the knowledge they gained in previous increments, both from developing the software and from users, to iterate the current version of the software during the development of current and future increments. The implementation starts with a simple set of requirements, and iteratively evolves until a full version of the software is implemented in a plan-driven way. The Rational Unified Process (RUP) is an example of a plan-driven incremental process (Jacobson et al. 1999). There are four phases of development in RUP, each of which is organized into a number of separate iterations that must satisfy defined criteria before the next phase is undertaken. Each iteration ends with a prototype of the software.

One category of iterative and incremental processes are agile processes that evolved to address the limitations of the plan-driven incremental approaches for software development (Beck 2001). The most widely used processes in software industry are XP and Scrum (Salah and Petrie 2009). Some of the major preferences in XP development are short development increments, simple design, pair programming, and customer collocation. The Scrum development process is described in subchapter 5.1.2.

Processes Used by IT Professionals

The results from a survey conducted in Iceland in 2009 (Larusdottir et al. 2009) show that almost half of the respondents use their own process, which is based on some predefined software development process, but adjusted to that particular software development project. About one third of the participants use the agile process Scrum and the remaining 20% use other processes, including for example the Waterfall process and Extreme Programming.
The results from a survey from Switzerland show that the most commonly used development process is also an internal process used by more than one third of the respondents and RUP was used by 20% (Vukelja et al. 2007). Agile methods were used by 15% of the respondents, and around 20% used no method.

The results from a survey study conducted in Norway (Bygstad et al. 2008) show that one third of the respondents used a defined software development process. Almost all the respondents, who did not use a defined development process, used some techniques and tools though. The majority of the respondents, who used a defined development process, used their own process and around 15% of them use an agile development process.

The results from a survey conducted in Sweden (Gulliksen et al. 2004) show that the most commonly used processes were an internal model and RUP, used by around one third of the respondents each. Seven percent of the respondents used agile processes at that time.

When the results from these studies are summarized, the most common way of planning the software development is using an own process. The agile processes are not that common in these studies. The highest number of participants using Scrum is in the Icelandic study. That could be because that is also the most recent study.

**5.1.2 The Agile Software Development Process Scrum**

Two studies in this thesis, which are described in papers III and IV, focus on the Scrum development process. In Scrum the projects are split up in two to four weeks long iterations called sprints, each ending up with a potentially shippable product. In Scrum self-organizing and well compounded teams are heavily emphasized, typically with six to eight interdisciplinary team members. One of the benefits of using agile development processes was claimed to be that customers’ needs are taken more into account than when developing software using sequential processes (Schwaber 1995). The Scrum process has been criticized for not involving real users in the software development and for not adequately addressing their usability needs (Singh 2008).

Scrum appeared as a software development process in 1995 (Sutherland 1995). Scrum is a simple process with a few ceremonies, roles
and artefacts. The roles are called Product Owner, Scrum Master and Team (Schwaber and Beedle 2002). The Scrum team is self-organising and works independently during the sprints. Daily Scrum meetings are prescribed where the Scrum team meets and plans the work during the day, and where the tasks are distributed in the group. The work in the Scrum team should be guided by collaboration and communication. Demos of the outcome are made at the end of every sprint.

The Product Owner has the responsibility to represent the needs and ideas of the customer. In this work, the Product Owner is responsible for writing the so called user stories and for managing these in a document called the Product Backlog. User stories are described in a standardised way to express the users’ needs and the business value in sentences with a specific structure. The Product Backlog contains the requirements for the software to be built often described as user stories. Parts of the Product Backlog may be technical, and parts may be more human oriented. The project is often analysed to some extent and a vision of the product is made before the actual iterations start. That period is often referred to as Sprint Zero or pre study phase.

The sprints are planned in sprint planning meetings, and the requirements to be addressed in each sprint are defined in the Sprint Backlog. By the end of each sprint, the Scrum team should have a potentially shippable product to release. At this point it is recommended in Scrum that the team collects feedback from the stakeholders to adapt the software to the users’ needs in subsequent sprints. The team and the Product Owner define an exit-criterion of the items in the Sprint Backlog called “the definition of done”. It describes when the items are completed and can be delivered to the customer. The definition might include descriptions of tests to be done. The Scrum board is a physical or electronic task board used by the team members to manage the different tasks and what to do during each sprint. The Scrum Master is often considered as a coach for the Scrum team. The Scrum Master is responsible for the process used during the software development and can for example decide how long the sprints are.

In Scrum the clients’ needs should be taken more into account than in older development processes, like Waterfall and RUP (Schwaber 1995). Through the Product Owner role, the priorities of the clients are defined.
That does not necessarily mean that the end-users’ needs are taken into account, because the client representatives are not always the ones who will use the software. The Product Owner does not always state the needs for usability of the software either (Singh 2008). One of the main conclusions in an extensive literature survey on the integration of the usability needs into agile processes is that the end user needs have not yet been sufficiently included in the agile development processes (Sohaib and Khan 2010).

5.1.3 Human-Centred Design Approach

The ISO 9241-210 standard (2010) (Human-centred design for interactive systems) is a common framework for including the users’ needs in software development. This is a recently re-worked version of the ISO 13407 (1999). The ISO 9241-210 standard (2010) defines the human-centred design as:

**Human-centred design:** Approach to system design and development that aims to make interactive systems more usable by focusing on the use of the system, applying human factors/ergonomics and usability knowledge and techniques (p. 2)

It is stated in the standard that:

*Human-centred design activities can be incorporated in design approaches as diverse as object-oriented, waterfall, HFI (human factors integration), agile and rapid development (p. 10).*

This human-centred design approach should therefore fit into the agile development process Scrum as well as in any other development processes.

Four major human-centred activities are described in the ISO 9241-210 (2010) standard that shall take place during the development of software. These are:

(a) Understand and specify the context of use,

(b) Specify the user requirements,

(c) Produce design solutions to meet these requirements,

(d) Evaluate the designs against requirements.

It is stated in the standard that these four human-centred activities are interdependent and each activity uses outputs from the other activities. This
means that when a specification of the context of use has been made, the user requirements should be described based on this understanding. Additionally, design solutions should be made based on these user requirement specifications. Evaluations should be made on the outcomes of each of the three human-centred design activities and the outcomes should be iterated according to the results of the evaluation, where appropriate. User centred evaluation therefore has a vital role in human-centred design.

An interview study with 12 usability practitioners was conducted in 2007 in Australia to examine their human-centred design activities (Bruno and Dick 2007). The main success factor reported in that study was that each human-centred design activity should be planned, conducted and the results analysed. Moreover, each human-centred design activity should be iterated as often as possible within that activity, combining evaluation in all the other activities. In one of the projects where the usability of the developed software was poor, the only human-centred design activity was evaluation. One of the main conclusions in the paper is that all four human-centred design activities need to be conducted and the outcomes iterated for the usability of the software to be reasonable. The participants reported that the main constraints for choosing a particular activity were the stage of the project, availability of IT professionals and time.

In papers III and IV it is studied how user centred evaluation, one of the human-centred design activities, is conducted in the Scrum process. In addition, the purpose of conducting evaluation is analysed in paper IV according the definition of these human-centred activities described in the ISO 9241-210 (2010) standard.

5.2 Related Work on the UCE Factors

In this subchapter the related work on the UCE factors: the evaluation methods used, the context of the evaluation, the purpose of conducting evaluation, the evaluators and the users, studied in papers III and IV will be described.

5.2.1 Evaluation Methods Used

Several studies have been done on how human-centred activities are conducted by IT professionals in practical settings. User centred evaluation
is included in several of those. The type of evaluation used in practical settings is studied in paper IV.

A descriptive online survey was conducted in Switzerland on how usability activities are included in software development (Vukelja et al. 2007). The respondents were mainly software developers and project managers. The results show that about 40% of the respondents conduct user testing, but less than 10% of testing results are used to do big changes of the software and 20% result in small changes. The majority of the respondents who do not conduct user testing consider user testing useful. One reason for this could be that less than 10% of the participants say that they had high HCI knowledge.

The results from an international survey show that the most commonly used usability method is heuristic evaluation, used by 70% of the respondents (Rosenbaum et al. 2000). Lab usability testing was conducted by 65% of the respondents, which was also the highest rated method out of the eleven methods included in the survey. The 134 respondents were mainly practitioners making software for various domains.

The practice of user performance measurements was examined in a survey study in 1993 in nine European countries (Dillon et al. 1994). The results show that 60% of the respondents did formal user performance measurements at that time and 75% did informal testing. When asked about how important usability evaluation was rated in the company around half of the respondents reported the answers “very important” or “essential” and less than 20% reported that it was regarded as “unimportant” or “not essential”.

5.2.2 The Context of Evaluation

In some research studies, results are reported on the context of the evaluation in terms of the project factors, client factors and the usability work factors described in subchapter 3.1. The frequency of conducting user centred evaluation in Scrum projects, which is one of the usability work factors, is studied in paper III.

The results from a survey and interview study conducted in Denmark show that some type of usability evaluation was conducted in almost 75% of the companies involved (Bak et al. 2008). The study did not
analyse what evaluation methods were used, and if the evaluation was analytical or empirical. A similar study was done in Italy (Ardito et al. 2011) and the results on the usage were similar; some evaluation was done in 72% of the companies involved. Internal evaluations were conducted in half of the companies involved in the study, but less than 20% conducted external evaluations by external consultants.

The major obstacles for doing user centred evaluation were examined in a study conducted in Denmark (Bak et al. 2008). The two main obstacles found were resource demands both in terms of time and money, and an obstacle called the “developer mind-set”. One example of issues in this category is that the respondents mentioned that they find it hard to think like users. In relation to this, some informants described that the main focus of IT professionals was on the programming aspect to write beautiful code and not so much on participating in a usability evaluation. In a similar study in Italy (Ardito et al. 2011) the most frequent obstacle mentioned was also resource demands. The most frequently mentioned advantage of usability evaluation was quality improvement, reported in almost half of the cases. In the study by Vredenburg et al. (2002) the main benefit for doing usability evaluation was that the practitioners gain understanding in the context of use whereas the weaknesses mentioned were high cost and versatility. The benefits and weaknesses of doing informal expert reviews and formal heuristic evaluation were similar, the benefits being the low cost and speed but the weakness being that users are not involved in the evaluation. In these three studies of the obstacles of conducting evaluation, the budget resources from the client is the major influencing factor on the evaluation activity.

5.2.3 The Purpose of Conducting Evaluation

Some studies have examined the purpose of the evaluation. More precisely, in what development stages evaluations are conducted and when the results are used has been examined in some of these studies. The main development stages are often described as: requirement analysis, design, implementation and testing, like the stages that are described in the simplest form of the Waterfall process (Royce 1970). In this subchapter an overview of the results from these studies is given.
The usage of 96 user experience evaluation methods in various software development activities was studied in recent study (Vermeeren et al. 2010). The methods were analysed according to at what stage in the development process the method could be used. Most of the methods could be used in the implementation and testing stages, and around one-third could be used either in requirements analysis or design stages. About one-third of the methods can provide quantitative data, one-third qualitative and one-third can provide both quantitative and qualitative data.

A survey study was conducted in Switzerland by Venturi et al. (2006). Qualitative evaluation was conducted at least once during the software development by almost 70% of the participants, most frequently during the design phase by almost half of the participants. Quantitative evaluation was used in at least one phase by around 40% participants, most frequently in the design phase and testing phase by around 20% each.

The use of field methods for designing and evaluating software was studied in an international web based survey (Monahan et al. 2008). The respondents were usability practitioners in education and industry. The results show that of the methods asked about, user testing was the most common method for evaluating the software, used in 70% of the cases. All the other methods were also used for evaluation, including observations, interviews, contextual design, probes and diaries. The purpose of user testing was mainly to evaluate, but user testing was also used to requirements gathering, understanding the context of use and for inspiring concept creation. The results show that more than half of the respondents rated observations as an extremely effective method and about 40% of the respondents rated user testing as extremely effective. The most influential factor for choosing a method for participants working in the software industry was time constrains.

A survey study was conducted in Norway, Finland and Sweden (Bark et al. 2006) where the HCI methods used were analysed according to the software development phase in which the methods were used. Seven methods were mentioned being used in the evaluation phase of the development. User testing was conducted by almost all the participants, expert evaluation by almost 90% and heuristic evaluation by 75%. Cognitive walkthrough was the method with the lowest usage, little less than 50%. The rating of the usefulness of the methods had almost the same
pattern as the usage; user testing was highest rated, followed by expert evaluations and heuristic evaluation. Surveys and cognitive walkthrough were rated as the least useful methods.

5.2.4 The Evaluators and Users

The person conducting evaluation can have various job titles. In paper IV the purpose of evaluation and the use of different evaluation methods are analysed according to the evaluator’s role in the software development. In paper III the role of the person conducting usability testing is analysed.

An international survey was conducted to study the activities of HCI professionals and software engineers (Jerome and Kazman 2005). The two groups of professionals did not agree on who conducted usability evaluations. Over 90% of the HCI professionals said that they themselves conducted the usability evaluation, and the remaining ones said that quality assurance professionals conduct the usability evaluation. Half of the software engineers said that quality assurance professionals handle usability evaluations, and one-third said that they themselves conducted the usability evaluation, but they do not mention that the HCI professionals conduct usability evaluations.

The results from a study from Norway show that over 90% of the participants conduct empirical evaluation (Bygstad et al. 2008). The respondents in this study were general managers and IT managers and they were asked to give answers for the whole company. One to ten users were included in the usability testing in around 75% of the cases and more than eleven users in the remaining of the cases. When asked about how the users were selected for the testing, 40% responded that it was a representative sample of users. These respondents gave user testing the highest rating. The respondents who asked arbitrary sample of users to participate in the evaluation and those who asked employees from the customer to participate gave user testing the lowest rating of importance. This could mean that user testing was not as effective for them as for the participants who asked representative sample of users to participate in the empirical evaluation.

Data on the role of the evaluators, their knowledge of usability in general and their knowledge of the type of interface evaluated should be gathered when conducting and researching user centred evaluation.
Additionally, data should be gathered on other UCE factors like the type of evaluation conducted, the context of the evaluation, the purpose of the evaluation and the participants in the evaluation, because all these UCE factors could influence the results of the evaluation.

5.3 Data Gathering and Analysis in Practical Settings

In this subchapter the data gathering conducted in papers III and IV and the analysis approach are described. Data were gathered using a questionnaire in paper III and interviews were conducted in papers III and IV.

5.3.1 Questionnaires

In paper III questionnaires were used to gather information on testing practices in software projects where the Scrum development process was used. Twenty five informants participated in a web based survey study containing 26 questions. There were five main sections in the questionnaire:

(a) Background and experience of the respondent.
(b) Information on the company, where the respondent works.
(c) The software development process used in the company.
(d) To which extent and who is conducting different testing techniques.
(e) The change in conducting software testing when compared with previous or parallel software development processes.

The respondents were all working full time on software development in eighteen organizations in Iceland. They developed software of various sizes for different purposes.

5.3.2 Semi Structured Interviews

Semi structured interviews were used in papers III and IV to gather information on how evaluation is conducted by IT professionals working on software development.

All the participants in the study described in paper III were asked if they wanted to take part in an interview study after the survey. Six
interviews were conducted for the pool of nine people that agreed to being interviewed. The purpose of the interviews was to collect richer data on usability and acceptance testing than was gathered in the survey. All the interviews were conducted face to face by the same person. The interviews were recorded, transcribed and were then analysed by another person.

In paper IV semi structured interviews were conducted with 21 participants in Sweden grouped in five different roles, which are: usability expert, interaction designer, business analyst, developer, and Scrum manager. The participants were working for fourteen various sized companies. The participants developed different types of software. All the participants were using the Scrum software development process.

An interview template was used containing questions related to the background of the participants, their experiences using the Scrum development process and their experiences conducting user centred evaluation. Questions were adapted in accordance with the organizational role of each informant. In almost all the interviews two researchers were present, one acted as the interviewer and another as a note taker. The interviews were around one hour long. The interviews were recorded and transcribed. The interviews were coded in the software program Atlas.ti for data analysis.

Data from the interviews were thoroughly analysed by two researchers working together do ensure quality. Thematic analysis was used as an analytic approach where data was reviewed, organised and analysed in order to identify themes or codes in an iterative manner (Ezzy 2002). First a scheme of detailed codes was made and a couple of interviews coded with that coding. Then the codes were made more general so that similar subjects would fall in the same code. The final version of the code scheme contained thirteen codes and all the interviews were coded according to those.

At this stage the two researchers who carried out the analysis discussed their findings and made interpretations, including search for alternative understandings and interpretations to statements made by the participants. The quality criteria defined by Klein and Myers (1999) were used to ensure quality in this work and in the subsequent writing up of the data.
5.4 Results

In this subchapter the goals of the studies in papers III and IV are described and the results are summarised.

5.4.1 Paper III: The Focus on Usability Testing

A survey study and an interview study were conducted and described in this paper. The goal of the survey study was to explore how testing is conducted and to what extent testing techniques are used by IT professionals in Iceland using Scrum as their development process. Moreover, the use of Scrum by the IT professionals was investigated. Various types of testing were explored. The more traditional ones, namely the unit, integration, system and acceptance testing, were compared with the non-functional testing techniques like usability, performance and security testing. Additionally, alpha and beta testing were explored. The goal of the interview study was to gain a better understanding of the data gathered in the survey focusing on usability testing and how it compares with acceptance testing.

The fundamental roles, activities and artefacts suggested in Scrum development process were used in nearly all the companies involved and the sprints were usually two to four weeks as recommended. A potentially shippable product after each sprint was only accomplished in around two-thirds of the cases through. This was less than expected, because this is one of the fundamental issues in the Scrum process.

About half of the respondents said that testing became easier in Scrum than when using other processes prior to using Scrum or when using other processes in parallel with using Scrum. The participants also said that more testing was done in Scrum than using the other processes. Only 12% said less testing was done. The reason could be that one team member has often the testing role in Scrum and the daily Scrum meetings give good overview of what needs to be tested.

Usability testing and performance testing were practiced in a similar way, but unit, integration, system and acceptance testing were much more frequent. However, only 4% said that they did not evaluate the usability, but 35% said that the extent was little. Some examples from the interviews show that IT professionals conducted formal usability testing typically once or twice a year. The participants were willing to do formal
evaluations such as performance measurements on extensive parts of the software more frequently, but formal evaluations did not fit into the project work because the iterations in Scrum are typically two to four weeks. The main reason for not conducting usability testing was lack of time. Examples from the interviews in the study show that the IT professionals would prefer to carefully plan their tests when the focus of testing was on attributes of quality like on usability, security and performance. Planning and conducting the tests requires extensive workload from the IT professionals, which is in contrast with the fundamental principles in Scrum: simplicity and speed.

5.4.2 Paper IV: Evaluation in Scrum Projects

This paper focuses on usability professionals’ experiences of integrating user centred evaluation into the Scrum development process. The overall research question of the paper is: How do the usability professionals apply the user-centred evaluation methods? The research question is examined in an in depth interview study. Moreover, the influence of the results from the user centred evaluation on the other three human-centred design activities is studied.

The following research questions are analysed in the context of Scrum projects in the paper:

(a) What methods do usability professionals use to evaluate their product?

(b) Does the evaluation methods usage vary among the different professional roles?

(c) What is the purpose of the evaluation?

The results show that the usability professionals used various user centred evaluation methods during the Scrum projects. The business analysts mainly used one approach though, which was asking users about their opinion where as the other roles used a combination of three or four approaches including observing users, asking users, measuring user performance, gathering feedback from user surrogates and conducting inspection evaluation.

The results show that 62% of the respondents observed users in some way. That could be when using paper prototypes in an informal
evaluation session and asking users to solve some tasks and observe them. Particularly the usability experts and interaction designers used this way of evaluating.

Almost all the participants used some informal qualitative methods like observing a few users, getting feedback on the usability from users through group meetings, blogs and chats. Additionally, analytical qualitative methods like peer reviewing were conducted by more than half of the participants.

Around 20% of the participants in the study measured users’ performance in a quantitative way. The measurements were conducted by some of the usability experts, one of the Scrum managers and one interaction designer. Quantitative measurements were not conducted by the software developers and business analysts. Many respondents mentioned that they only had time for thorough user centred evaluation once a year.

The main contribution in the paper is an understanding of how the Scrum context changes the conditions for user centred evaluation for the IT professionals. The lessons learned and recommendations to IT professionals of how to maintain the user centred focus and how user centred evaluation methods can be integrated into the Scrum process are given in the paper. Moreover, recommendations to HCI researchers on what to focus their research studies on are also given.

5.5 Discussion

In this subchapter the results in papers III and IV are discussed and compared to related work.

The main findings in paper III show that unit, functional, system and acceptance testing were conducted to a wide extent. Usability, security, performance, alpha and beta testing were much less frequent. Still only 4% said that they did not evaluate the usability, but 35% said that the extent was little. These numbers are similar to the ones in the study from Norway, were over 90% of the participants test the usability (Bygstad et al. 2008). The respondents in the Norwegian study were IT managers and in paper III the respondents were IT professionals responsible for testing. In both cases the respondents were asked to answer for the whole company. The results from a Danish study show that around 75% of the companies involved in that study evaluated the usability (Bak et al. 2008). Similar results are
shown in an Italian study (Ardito et al. 2011). So the results from paper III show higher quantity of usability evaluation conducted than in the Danish and the Italian study. The explanation could be that, in paper III, respondents could give the alternative of doing evaluation infrequently, so if they had done it once, even several years ago, they could give a positive answer, where as in the Danish and Italian studies the respondents were not asked about how often they conducted evaluation. Also in the Danish and Italian questionnaires, usability evaluation was explained by using methods to evaluate, but in paper III it was explained more like the activity of testing the usability and the use of evaluation methods was not emphasised.

In what way the IT professionals conducted user centred evaluation was examined in the study described in paper IV. The results show that around 20% of the participants in the study measured users’ performance in a quantitative way. The results from a study by Venturi et al. (2006) show that quantitative evaluation was done by about 40% of the participants, which is higher than in paper IV. The results from an international survey by Rosenbaum et al. (2000) show that 65% had done lab usability testing. When the results from paper IV are compared with the results from the studies mentioned above, it can be stated that less quantitative evaluation is done in 2011 using the development process Scrum, than was done overall in software development in the other studies. The reason could be that quantitative evaluation is not as widely used today as ten years ago generally in software development, or it could mean that less quantitative evaluation is conducted in Scrum project than in other software project where other processes are used.

The results from paper IV show that 62% of the respondents observed users in some way. The results from previous studies give a broad variance in this aspect. In the study from Norway over 90% of the participants conducted user testing (Bygstad et al. 2008) and in the study by Monahan et al. (2008) 70% of the participants did user testing. In a study from Switzerland (Vukelja et al. 2007) about 40% of the participants conducted user testing, and in an international study by Vredenburg et al. (2002) similar results are described. It is therefore hard to say if there is some trend in these studies.

The results from the study described in paper IV show that various informal ways of evaluation were used by IT professionals. Moreover,
Analytical qualitative methods like peer reviewing were conducted by more than half of the participants. This is slightly higher than the results from the study conducted in 2002 where 30% of the participants had done informal expert reviews and it was not at all highly rated by them (Vredenburg et al. 2002).

Many of the respondents in the study described in paper III said that user centred evaluation could only be done once or twice a year. These results are similar to the results from paper IV. One of the main obstacles for doing usability evaluation was resource demands both in terms of money and time in a Danish study (Bak et al. 2008) and in an Italian study (Ardito et al. 2011). In those studies the results were not analysed according to the software development process used. Time constrains was the most influential factor for choosing a method for participants working in on software development in an international study by Monahan et al. (2008).

5.6 Lessons Learned

The second research question is: How do IT professionals conduct user centred evaluation in the Scrum development process? The results described in this chapter indicate a possible answer to this research question. It can be concluded that usability testing is not as frequently conducted as acceptance testing. IT professionals using the Scrum development process conduct user centred evaluation in an informal way. They evaluate the software with real users or with user surrogates, and by conducting analytical qualitative evaluation. Not many IT professionals conduct empirical quantitative user centred evaluation. The main reason given for not conducting user centred evaluation more often is lack of time.
6 Contribution

The research topic of this thesis is user centred evaluation, which is explored by comparing the effectiveness of using different evaluation methods and by studying how user centred evaluation is conducted by IT professionals using the software development process Scrum. The main contribution of this thesis is summarised in the following four statements.

**Empirical evaluation is often conducted**

Many participants in the study described in paper IV obtain feedback from users or user surrogates. That is positive, because the results from paper I show that the empirical method think-aloud is more effective in finding real problems than the analytical methods heuristic evaluation and cognitive walkthrough. By using the heuristic evaluation and the think-aloud method similar quantity of problems were found, but around 25% of the problems found by using heuristic evaluation were false problems. The results from using the think-aloud method are therefore more reliable.

**Informal qualitative evaluation is often conducted**

The results from paper IV show that usability professionals use many informal qualitative methods to evaluate their software using the Scrum development process. The results show that informal evaluation is conducted to get feedback from users and to get feedback from IT professionals by peer review and with expert evaluation. The main constraint for conducting user centred evaluation reported in papers III and IV is lack of time. The IT professionals find solutions to that by conducting evaluations in an informal way.

**Quantitative evaluation is rarely conducted**

The results from papers III and IV show that user performance is rarely measured by the IT professionals. Additionally, only a few participants use surveys to gather quantitative data. By gathering quantitative data a concrete measurement of the user performance and other factors is conducted. It is positive for the IT professionals to know exactly the state of the usability and user experience of the software. The results of
quantitative evaluation could be used to measure if a particular goal for the usability of the software has been reached.

The reason for not conducting quantitative evaluation is that in Scrum the emphasis is on short intervals between each delivery of workable software to the customer and the IT professionals do not have time to gather quantitative data with many participants. Additionally, because of the many deliveries there is no natural timing for doing thorough user performance measurements.

Using a combination of methods is effective

In paper IV the results show that the participants use various evaluation methods during their software projects. The results from paper I show that using the heuristic evaluation before conducting think-aloud evaluations extends the effectiveness of using the think-aloud method.

Moreover, a model was developed called the UCE model to illustrate a point of reference when describing and studying user centred evaluation. The factors involved when conducting evaluation are illustrated and described in the UCE model, which can be used to design future research studies on user centred evaluation. In addition, it can be used by IT professionals to explain the evaluation conducted in a systematic way. The context of user centred evaluation was also illustrated and described briefly.
7 Future work

After summarising the results in this thesis, new ideas appear for studies to extend the knowledge in the area even further. In this chapter, five questions are explained that could be examined by HCI researchers in future research studies on user centred evaluation. The questions are based on the results in this thesis.

*How can new evaluation methods based on the development process Scrum be used in an efficient way?*

New user centred evaluation methods that fit into the Scrum process are needed. There are various alternatives, three of which will be discussed in the following.

The first alternative is to suggest methods that use the current artefacts in Scrum as a basis for evaluation, like using user stories as a basis for defining tasks for empirical evaluation. The second alternative is to suggest new methods where the emphasis is to evaluate various quality factors, like safety, performance and usability at the same time. The results from paper III show that these quality factors are not often prioritized when testing the software. Evaluating more than one factor in the same evaluation could make the evaluation more efficient. The third alternative is to suggest methods that are low cost methods or low effort methods, meaning that the IT professionals get valuable results, while using a reasonable amount of time. The results in paper IV show that the IT professionals choose informal qualitative methods for evaluating the usability of the software they are developing because these can be used rather quickly and give valuable results.

These new methods could be studied in experimental settings asking computer science students to take part in the research studies. Half of them could use a new method and half of them could use more established methods like the think-aloud method. The results could be compared and the evaluators’ satisfaction measured. When the new method has been modified according to these studies, it could be presented to IT professionals and examined in pilot projects in software development using the Scrum process.
What are the benefits and weaknesses of conducting qualitative evaluation in Scrum projects?

The results in this thesis show that various qualitative evaluation methods are used by IT professionals using the development process Scrum. These methods should be studied in more detail to extend the knowledge of the benefits and weaknesses of these methods and where these fit in the Scrum development process. It has also been pointed out by Law (2011) that qualitative evaluation methods seem to be “the winners” when evaluating user experience and should be studied further.

The effect of combining the use of various evaluation methods, both qualitative and quantitative, during a software development project has not been explored much, as pointed out by Hornbæk (2009). The results from paper I show that combining the use of two methods can be effective. More precisely, using the heuristic evaluation before conducting think-aloud evaluations extends the effectiveness of using the think-aloud method. The results from paper IV show that IT professionals use various user centred evaluation methods during their project work. Therefore researchers in the field are encouraged to study what effect it has to apply various ways of conducting user centred evaluation, both qualitative and quantitative.

Are the results from user centred evaluation consistent, complete and valid?

As pointed out by Hartson et al. (2001) and described in subchapter 4.2.2, evaluators want reliability, meaning that the user centred evaluation results are consistent. In addition, evaluators want thoroughness, meaning that they want the results to be complete, and they want the results to be valid, meaning that real problems are captured in the evaluation.

To extend our understanding of whether using evaluation methods gives information on the problems users actually have when using software in real life, researchers could contact the users that participated in user centred evaluation of a particular version of the software before installation and obtain feedback from the users after a period of actual use. The goal would be to check if the users have had the usability problems while using the software that were registered in an evaluation and were not corrected. Another possibility is to watch users or video record their actual use of
particular software for some period of time and compare those results to results from user centred evaluation.

**How useful is the UCE model?**

The usefulness of the UCE model for researchers and IT professionals could be studied, and the UCE model could be modified according to the results. The UCE model could be illustrated on a website, giving access to relevant information on the factors.

Researchers could use the UCE model to describe their experimental design, if comparing some factors in the UCE model, to illustrate the goal of their research. Researchers could study in more detail the combination of some of the factors in the UCE model to gain more knowledge on how the factors correlate to one another. For example, one might study if difference in age of the participants in user observations does affect the results of user centred evaluation of software from different domains.

Moreover, the UCE model could be used both by researchers and IT professionals to describe how a particular user centred evaluation was conducted. The status of each of the factors should be described to get an overview of what was done and how the evaluation was conducted. A database to collect experiences of using the various combinations could be made so researchers and IT professionals could gather knowledge about how effective the evaluation was in each particular case.

**Is feedback gathering more appropriate as a term than evaluation?**

In paper IV the results show that IT professionals rarely name the particular user centred evaluation methods they used. Additionally, they talk about gathering feedback from users and do not name it evaluation.

The definition in the ISO 9241-210 (2010) standard on user centred evaluation is really focusing on the feedback gathered through evaluation from the users’ perspective. In the 9241-210 the three main outcomes from user centred evaluation are evaluation results, conformance test results and long-term monitoring results. Asking IT professionals how they review their software and how they gather feedback from the users’ perspective could give results that are different from asking the IT professionals about how they conduct user centred evaluation.
8 Conclusion

In this thesis the topic user centred evaluation is examined in experimental and practical settings.

The effect of using various evaluation methods was examined in the experimental settings in a quantitative way. By using the think-aloud method and the heuristic evaluation similar quantity of usability problems were found. Only about half of the quantity of problems found using these methods, were found using the cognitive walkthrough method. When using heuristic evaluation many usability problems, registered by the evaluators did not occur as problems when users were using the software. This result was confirmed in another study described in this thesis. Using heuristic evaluation in advance of think-aloud evaluations did improve the results of the think-aloud evaluations, especially in finding critical problems. Based on these results IT professionals are encouraged to use the think-aloud method whenever possible. Preferably IT professionals should use a combination of the heuristic evaluation and the think-aloud method to make their evaluations more effective. Researchers are encouraged to examine the effect of using more than one method for evaluating software.

The use of user centred evaluation methods by IT professionals using the software development process Scrum was examined in two studies in practical settings. The results show that many IT professionals conduct informal qualitative evaluation with users, user surrogates or other IT professionals. Quantitative user centred evaluations are conducted by few participants and infrequently. Based on these results IT professionals are encouraged to evaluate their software including users in an informal way frequently rather than waiting for the right time to do thorough quantitative evaluation. Researchers are encouraged to focus on qualitative evaluation to inform the IT professionals of possible ways of improving this type of evaluation.

Various factors of user centred evaluation can be studied in experimental settings and compared in various ways. Because IT professionals working on software development are pressed by time schedules and it is almost impossible for them to take part in such experiments, asking computer science students to take part in experiments is often the only possibility. The students do not have much experience in
evaluating nor do they have much knowledge of the software being evaluated. Moreover, these projects are rarely run in the whole context of evaluation, taking all constrains and other project activities into account, like budgetary constrains. These limitations could affect the results of the studies.

In practical settings IT practitioners can be asked about how they conduct user centred evaluation in their software development. Some of the limitations of asking IT professionals are that they could misunderstand the questions, use different terms than the researcher when talking about a particular subject and find it hard to explain how they do a particular activity. Additionally, they could have forgotten how a particular activity was conducted.

Researching user centred evaluation in experimental and practical settings gives valuable knowledge for IT professionals and researchers on what to focus their future activities on taking the limitations of the research studies into considerations. In this thesis, knowledge is gained from research studies in experimental settings on how much a variation in one factor of user centred evaluation affects the other factors. Moreover, knowledge is gained from the research studies in practical settings on how the process of user centred evaluation could be improved. The interplay between the results from research studies in experimental and practical settings on user centred evaluations is vital to extend the knowledge on this research topic further.

Thereby, I recommended to HCI researchers to seek knowledge on user centred evaluation both in experimental and practical settings.
9 References


Cockton, G., 2005. I can’t get no iteration. Interfaces, 63.


Sutherland, J., 1995. Business object design and implementation workshop. In: Addendum to the proceedings of the OOPSLA 1995 on object oriented programming system, languages and applications, Austin, Texas, USA.


10 Appendix I: Other Publication

This section lists my other publication, not included in this thesis. The list is chronological, with the most recent publication at the top.


