Usability Evaluation Methods

The Reliability and Usage of
Cognitive Walkthrough and Usability Test

Niels Ebbe Jacobsen
Department of Psychology
University of Copenhagen
Denmark

Ph.D. Thesis
October 5, 1999

Table of Contents

Preface

This thesis is submitted in fulfillment of the requirements for the Ph.D. degree at the Faculty of Humanities, Department of Psychology at the University of Copenhagen, Denmark. The thesis includes this 98-pages summary and appendices containing a paper submitted for journal publication (Jacobsen & John, 1999), three published conference papers (Jacobsen, Hertzum & John, 1998a; Jacobsen, Hertzum & John, 1998b; Hertzum & Jacobsen, 1999), and an unpublished manuscript (Hertzum & Jacobsen, 1998).

Contents

Our[1] contribution to the field of human-computer interaction (HCI) has been centered on evaluating aspects of two usability evaluation methods (UEMs), namely the cognitive walkthrough and the usability test. Chapter 1 in this summary is devoted to an introduction to HCI with particular focus on UEMs, while chapter 2 contains a description of the two methods investigated (cognitive walkthrough and usability test). The most extensive study was the investigation of the learning process and the first time usage of cognitive walkthrough; this work is described in chapter 3. While working on this case we began studying the effect of different evaluators analyzing the same interface with the same UEM, the so-called evaluator effect. We have studied the evaluator effect both in cognitive walkthrough and usability test. In chapter 4 we describe the evaluator effect in cognitive walkthrough. In chapter 5 we describe the evaluator effect in usability test. The last chapter, chapter 6, discusses and draws perspectives on the work described in chapters 3, 4, and 5.

Four of the five appendices are included in this thesis submission in order to document earlier reports on the evaluator effect studies. The study about the evaluator effect in cognitive walkthrough has previously been reported in Hertzum & Jacobsen (1998) and Hertzum & Jacobsen (1999), and is included in appendices 2 and 3 respectively. The study about the evaluator effect in usability test has previously been reported in Jacobsen, Hertzum & John (1998a) and Jacobsen, Hertzum & John (1998b), and are included in appendices 4 and 5 respectively. As the early reports on the evaluator effect were constrained with regard to number of pages, I have extended the descriptions in chapters 4 (the evaluator effect in cognitive walkthrough) and 5 (the evaluator effect in usability test) in this summary. Hence, reading these two chapters would be sufficient to understand the main contents of the previous four reports. In addition, chapters 4 and 5 offer some new analyses that have not been revealed in the earlier reports. Opposed to the somewhat repeated reports in appendices 2, 3, 4, and 5, appendix 1 (Jacobsen & John, 1999) contains an extensive description of a case study, which is not repeated in the summary. Hence, chapter 3 introduces learning and usage of cognitive walkthrough, but only as a prelude to reading the case study paper in appendix 1. After reading appendix 1, the reader is expected to continue reading the remaining chapters 4, 5, and 6 in this summary. The explanation for this somewhat confusing structure is that I have insisted on submitting a thesis where *my* contribution (this summary) is separate from *our* (Morten Hertzum, Niels Ebbe Jacobsen, and Bonnie E . John's) contributions that are gathered in papers, articles, and reports in the appendices.

[1] Throughout this thesis I will use plural form to indicate what "we" have done. Although I am single-hander on this summary, the description of my work to a large extent covers what I have done in close cooperation with other researchers. Therefore, I have preferred to use "we" rather than "I" and "us" rather than "me". In a few places, however, I have chosen to use singular form, especially when the points raised are my own experiences or my own opinions rather than others.

Framing our field of interest

Our work was intended to be truly interdisciplinary with strong connections to psychology and computer science. Starting with my own background and Ph.D. studies during the last three years is probably the best way to describe my interdisciplinary interest and how it has formed and developed over the years. Five years ago I earned a masters degree from the Department of Computer Science, University of Copenhagen, with a minors degree in information psychology from the Department of Psychology, also University of Copenhagen. During my masters thesis writing, I had the fortune to be supervised by at that time Ph.D. student Morten Hertzum (a computer scientist currently employed as a Senior Scientist at Risø National Laboratories) whom I have been working with throughout my studies. For one and a half years I worked in the industry as a systems developer, before I began my Ph.D. studies at Department of Psychology, University of Copenhagen. I have had the pleasure to be supervised by Associate Professor Anker Helms Jørgensen (a computer scientist with many years of employment in Department of Psychology, currently employed at the IT University in Copenhagen). Only half a year after my enrollment in the Department of Psychology, I found myself in a cubicle area of one of the first established Human-Computer Interaction Institutes (HCII) in the world. Associate Professor Bonnie John (a mechanical engineer with a Ph.D. in psychology) hosted me for one and a half years as a visiting research associate in HCII at Carnegie Mellon University, Pittsburgh, US. The last year of my Ph.D. studies I was back in Department of Psychology in Copenhagen to complete my projects.

Thus, the fields of computer science and psychology have certainly been important in my studies during the past ten years. However, the current work is neither psychology, nor computer science. Our work is in the field of *Human-Computer Interaction* (HCI), a multidisciplinary field with links to especially psychology, computer science, sociology, engineering, human factors, and design. The HCI field has its own conferences, its own journals, and its own institutes. Our background for understanding HCI and researching in this field, however, is in psychology and computer science. We believe that new achievements in HCI are to be found not in the parent disciplines but on the bridges between parent disciplines, e.g., the bridge between psychology and computer science.

Acknowledgements

This work could not have been completed without help and collaboration from colleagues, friends, and family. A great debt is owed to the three people that I have had the pleasure to work closest together with: Anker Helms Jørgensen, Bonnie E. John, and Morten Hertzum. I am, of course, pleased with the papers we wrote together, but I am especially grateful for the inspiring processes and discussions that we have had; activities in which I have acted as the naïve student and they (at least to me) have been the insightful teachers.

The reports and papers I have had the pleasure to write with others have been particularly important to me. Appendix 1 has been written in close cooperation with Bonnie E. John (Jacobsen & John, 1999). She ran the studies and collected the data material. I was invited to analyze and write a paper on the collected data. Although I have been the prime researcher and author on this study it could not have been compiled without persistent and enormously qualified supervision under Bonnie E. John. She has taught me much about investigating UEMs in general, but most importantly she has taught me the importance of paying attention to details. The weekly meetings with Bonnie, with this case study being the top-priority on the agenda, has been an invaluable part of my learning process; an experience that I will greatly benefit from in the future.

In this summary, the majority of the content in chapter 4 has been produced as a result of close cooperation with Morten Hertzum. Being the first author of our papers in appendices 2 and 3

(Hertzum & Jacobsen, 1998; Hertzum & Jacobsen, 1999) Morten Hertzum initiated and ran the studies of the evaluator effect cognitive walkthrough study. I was invited to participate in the analysis of the data and writing of the papers. In these studies Morten wrote the first drafts (based on analyses that we both compiled equally) after which we iterated on the papers tens of times until we both were satisfied with the end-result. Working with Morten has not only been extremely valuable from a professional point of view, but more importantly, our cooperation is a unique combination of effectiveness, efficiency, and enjoyment. I am grateful for having experienced this pleasant spirit of work.

The last project described in this summary (chapter 5) and in appendices 4 and 5 is a result of a cooperation between Morten Hertzum, Bonnie E. John and myself. Part of the data material was initiated and collected by Bonnie E. John (the usability test sessions). Having initiated two projects – one with Morten and one with Bonnie – both about the evaluator effect in usability test, I found it valuable to combine our efforts. I managed the project, and Bonnie was the eagle-eyed reviewer and commentator throughout the phases, while Morten and I compiled much of the analyses in close cooperation. The writing of the papers was initially my responsibility but almost all drafts were reviewed, commented, and corrected by Morten and Bonnie. Moreover, in some of the later iterations both Bonnie and Morten were (in turn) responsible for the drafts, which improved the quality of the papers significantly. However, I had the final responsibility of the contents of the papers before submission.

In the first half year of my Ph.D. period I conducted two preliminary studies, which I have reported in unpublished manuscripts (Jacobsen, 1996; Jacobsen 1997). These studies inspired me to continue further research as described in this thesis, but due to the limited data material of the preliminary studies, I have chosen not to include them here. Quite recently, I have had the pleasure to write some opinion papers and conference contributions with my advisor, Anker Helms Jørgensen, regarding epistemology in the field of UEM (Jacobsen & Jørgensen, 1998; Jørgensen & Jacobsen, 1998; Jacobsen & Jørgensen, 1999). These contributions are indeed important and interesting, but also fairly tentative in their contents and presentation. Although I have chosen not to submit these papers as a part of my thesis, I will thank Anker for the insightful discussions we have had during our work. I hope and believe that we will continue our work on epistemology in our chase for a proper theory-of-science model of the evolvement of UEMs.

Several people have helped me in commenting earlier drafts of this summary and helped us proofreading reports in the appendices: Chris Connors, David Crow, Julia G. Deems, Ejlif E. Jacobsen, Torben Elgaard Jensen, Yannick Lallement, Sofia Magno, Morten Schultz, Dave Winkler, and Anette Zobbe. Thank you all! I am also very grateful to Ejlif E. Jacobsen, Benny Karpatschof, and Thomas Scheike for their help on statistical issues in the evaluator effect studies. Many people have had impact on my work solely through discussions, in meetings, at conferences, and through e-mail writings. I especially want to thank: David W. Biers, Elaine Gilman, Wayne Gray, Morten Borup Harning, David Hilbert, Jacob E. H. Jacobsen, James R. Lewis, Rolf Molich, David Redmiles, Herbert A. Simon, Robert A. Virzi, and Cathleen Wharton.

Finally, I admire and I am deeply grateful to Sofia Magno who has had such great patience with me in my attempts to be on time with paper and thesis submissions; she also willingly acted as the last eagle-eyed proofreader on my work, for which I am truly thankful.

The Danish Research Councils supported my work. The views and conclusions contained in my work are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of The Danish Research Councils.

Abstract

Information technology that may work from a technological perspective is often perceived as being difficult to use. We forget how to program our VCR, we loose data on our personal computer because of inappropriate design, and more seriously, airplanes crash because of usability problems in cockpits or air traffic control towers (Reason, 1990). One way to improve usability of technology is to evaluate the design of technology before it is released. Usability evaluation methods (UEMs) applied in iterative development projects (Bailey, 1993) aim at identifying problems in the interaction between humans and systems in order to correct the problems identified.

We have focused on two UEMs, usability test (Lewis, 1982) and cognitive walkthrough (CW; Lewis et al., 1990). A usability test enables an evaluator to detect usability problems based on observations of users' interaction with a prototype of a system. The usability test has been applied in the industry and is accepted as the golden standard in usability evaluation. CW – based on a cognitive theory – enables an evaluator to make a detailed simulation of a fictive user's behavior in order to identify problems related to ease of learning. The result of using UEMs is a list of usability problems that should be communicated to systems developers in order to correct the problems identified in the system.

Several studies have compared different UEMs quantitatively in order to inform practitioners about the relative usefulness of the methods. These studies have revealed and compared the number of problems identified using various UEMs. We have found three problems with such comparative studies. First, they do not reveal *why* some UEMs detect more problems than others or *what* can be done to improve the UEMs. Second, in all comparative studies, different evaluators used different UEMs on the same system (between-subjects studies). Hence, we do not know if different UEMs or different evaluators using UEMs cause the deviating results of the comparative studies. Third, the result of the usability test has been thought to be independent of the evaluator. This assumption, however, has never been tested empirically.

We wanted to go beyond the comparative studies of UEMs to investigate the learning and usage of CW qualitatively. We wanted to answer questions about how CW is learned, how it is used, and to propose possible improvements to the method. Moreover, we investigated the predictive power of CW, i.e., how do problems predicted by evaluators using CW based on a specification document match problems found in a usability test based on a running system. We also wanted to investigate one reliability aspect of the usability test and CW, the *evaluator effect*. The question is if different evaluators analyzing the same system with the same UEM will identify the same problems.

In the qualitative study on learning and using CW we found the method easy to use and apply to a complex interface. However, the evaluators detected very different numbers and types of problems, and the predictive power of CW was disappointingly poor. Based on these results we have suggested three changes to CW to improve its accuracy and two changes to improve its reliability. Further, we recommend developing a computer-based tool to support the CW analysis in order to reduce the tedium of using CW and to integrate our suggested improvements to CW in the tool.

In the quantitative studies on the evaluator effect in CW and usability test we found substantial individual differences among evaluators. The study on the evaluator effect in usability test revealed that only 20% of the total of 93 usability problems were detected by all four evaluators, while almost half of the total problems were detected by not more than one evaluator. In comparison, the other study on the evaluator effect in CW revealed that no single problem was identified by all 11 evaluators and of the total of 33 problems collectively detected by all evaluators, more than half was detected by not more than one evaluator. These results suggest that none of the methods are as reliable as we thought. We hypothesize about causes for these findings and suggest future studies on UEMs.

Resumé (Abstract in Danish)

Informationsteknologi som måske fungerer rent teknisk anses ofte for at være vanskelig at bruge. Vi glemmer hvordan vi programmerer vores videomaskine, vi mister data på vores PC pga. dårligt design, men det er selvfølgelig langt mere alvorligt, at fly styrter ned, fordi der er problemer med at bruge teknologien i cockpits og kontroltårne (Reason, 1990). Man kan imidlertid forbedre brugen af teknologien ved at evaluere designet af teknologien, før den anvendes i en virkelig brugssituation. Evalueringsmetoder (UEMer = Usability Evaluation Methods) anvendt i en iterativ udviklingsproces (Bailey, 1993) har til formål at identificere problemer i samspillet mellem mennesker og maskiner med det overordnede formål at rette de identificerede problemer.

Vi har fokuseret på to UEMer: Usability test (Lewis, 1982) og cognitive walkthrough (CW; Lewis et al., 1990). En evaluator som anvender usability testen kan identificere problemer baseret på observation af brugerers interaktion med en prototype af et system. Usability testen har været anvendt i praksis og er generelt blevet anset for at være den gyldne standard indenfor evaluering af brugsvenlighed. CW er baseret på en kognitionspsykologisk teori (CE+). CW gør det muligt for en evaluator at gennemføre en detaljeret simulering af en fiktiv brugers interaktion med et system med henblik på at identificere indlæringsproblemer i brugen af systemet. UEMer finder altså problemer mht. brugsvenlighed og de kommunikeres videre til systemudviklere, der så forventes at rette de identificerede problemer.

Adskillige undersøgelser har sammenlignet UEMer kvantitativt for at informere praktikere om de forskellige UEMers relative anvendelighed. Disse studier har rapporteret og sammenlignet antallet af problemer identificeret ved brug af forskellige UEMer. Vi mener, at disse undersøgelser er problematiske af følgende grunde. For det første beskriver de sammenlignende undersøgelser ikke, *hvorfor* visse UEMer identificerer flere problemer end andre UEMer. For det andet har sammenlignende undersøgelser anvendt den samme problematiske metode, hvor de har brugt forskellige evaluatorer til at vurdere forskellige UEMer (i såkaldte ”between-subjekt” undersøgelser). Af denne grund ved vi ikke om resultaterne af disse undersøgelser er en effekt af forskel mellem de pågældende UEMer eller forskel mellem evaluatorer. For det tredje har man antaget at usability testen er pålidelig mht. hvem, der observerer brugerne. Denne antagelse er dog aldrig blevet undersøgt empirisk.

Vi ønskede at undersøge og forstå hvordan evaluatorer lærer og bruger CW for på baggrund af dette at komme med specifikke forslag til forbedringer af CW. Desuden ønskede vi at undersøge, i hvilken grad CW gennemført tidligt i udviklingsprocessen kan forudsige, hvilke problemer der bliver fundet i en usability test på et kørende system. Vi ønskede også at undersøge et aspekt af pålideligheden af de to teknikker, et aspekt vi har kaldt *evaluator-effekten*. Vil forskellige evaluatorer, som analyserer det samme system med den samme UEM identificere de samme problemer?

I det kvalitative studium fandt vi, at CW er let at lære og let at bruge selv ved evaluering af et komplekst system. Til gengæld fandt vi også, at forskellige evaluatorer identificerede meget forskellige antal og typer af problemer, samt at CWs evne til at forudsige problemer var skuffende dårlig. Baseret på disse resultater har vi foreslået tre ændringer til CW for at forbedre præcisionen af resultaterne samt to ændringer for at forbedre pålideligheden. Desuden har vi anbefalet, at der udvikles et computerbaseret redskab til at støtte en CW-evaluering og vores anbefalinger kan således implementeres i et samlet hele.

I den ene kvantitative undersøgelse af usability testen blev kun 20% af de i alt 93 problemer identificeret af alle evaluatorer. Næsten halvdelen af problemerne blev ikke identificeret af mere end én evaluator. Tilsvarende blev der i den anden undersøgelse af evaluator-effekten i CW fundet store forskelle mellem evaluatorer. Ud af 11 evaluatorers analyser fandt vi ikke et eneste problem, som alle evaluatorer havde identificeret. Langt over halvdelen af de i alt 33 problemer blev ikke identificeret af mere end én evaluator. Disse resultater indikerer at ingen af de to metoder er nær så pålidelige, som vi oprindelig troede. Vi har derfor søgt årsagerne til disse slående resultater og har på den baggrund givet forslag til undersøgelser som bør gennemføres.

1 Introduction

Usability evaluation methods (UEMs), which are our field of interest, have been found valuable for various reasons. In order to understand our motivation for investigating aspects of UEMs, we will introduce this summary by describing what usability is and why it is important, what has been done to improve usability of systems, and what UEMs are. Against this background we will be able to describe our specific motivation for investigating certain aspects of UEMs and describe our precise research questions. This introduction ends with an overview over the contents of this thesis and a reading guide.

1.1 Background

Conducting research in usability of technological systems is a fairly new discipline. Part of the reason is that information technology in the past 30 years has taken rapid steps towards changing our lives both in work settings and in our private sphere. Technology is developed to be used by the masses and much of the developed technology is not as easy to use as it should be. In the first sub-section we describe what usability is and why it is important to improve and ensure usability. In the second sub-section we have focussed on three important approaches to ensure and improve usability of technology. The first two approaches are development of specific *methods* and *models* to ensure and improve usability, namely the development of usability guidelines, and the development of formal, psychological models simulating a specific type of users. The third approach is related to practice as it focuses on the *process* of systems development as means for ensuring and improving usability of systems.

1.1.1 Usability: what is it and why is it important?

Our field of interest is usability of information technological systems. Information technological systems are any electronic system that collects, processes, stores, or distributes information, e.g., a desktop computer, an ATM machine, an airplane cockpit, a car stereo, a photocopy machine, a digital watch, a Virtual Reality game, a cellular phone, etc. An information technological system (hereafter just termed *system*) is developed for some specific or some more general purpose. In the traditional sense, systems have been developed to support people solving tasks in work-settings. However, systems also appear in other contexts not related to work-settings, e.g., in private homes and in educational settings. Thus, usability exists in some kind of environment framing a situation in which people are accomplishing a task using some system. This description has been advocated by Shackel (1991) among others (see Figure 1). The two UEMs we have investigated produce problem lists based on tasks being solved by people in a particular environment. Thus, Figure 1 frames our point of interest nicely. Before getting closer to defining usability, let us look at reasons to why there seem to be usability problems in these kinds of situations.

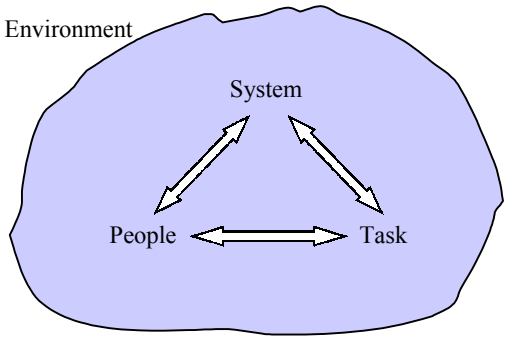

Figure 1. The building blocks of our interest: environment framing a situation where people solve tasks using some system.

In the early years, systems were developed to solve some very specific tasks for a small group of people. Systems were extremely expensive to develop, and they were often built, used and further developed by the same highly educated specialists. Usability of these systems was no prime issue, partly because the developers also used the systems (hence, they knew how to use them), partly because making the systems workable and bug-free was in itself difficult, leaving little attention to usability considerations. In the late sixties and early seventies, the first companies – for example banks and insurance companies – invested in systems dedicated to solving domain specific tasks, e.g., replacing manual account systems by electronic account systems. Suddenly, users did not have the benefit of knowing how the systems were built, as the developers and users were essentially two different groups of people. Among other problems these companies experienced usability problems, i.e., users had problems using the systems either because the system did not support the task to be solved by the user or because the functions to support the task in question were difficult to use. Although lessons were learned from early systems development, these lessons were not sufficiently communicated and where perhaps not sufficiently general to solve the problems of the eighties, where many more companies invested in information technology. In that decade some electronic appliances also found their way into private homes. Meanwhile, the development of technology moved faster than our understanding of the usage situation, which introduced even more usability problems. The nineties are known as the decade where the personal computer was introduced in private homes. Now, a few months from the millennium, nearly all Danish companies have computers, more than half of the Danish population has Internet access, and we are surrounded by electronic devices at work and in our private homes. Unfortunately, we are surrounded by many systems that we have great difficulty operating or that we do not know how to operate at all.

For example, in my everyday life, I am annoyed that I continuously hit the power off switch on my remote control when I wish to zap between TV channels, that the elevator does not close the door immediately unless I remember to keep pressing the button for the floor number for some seconds, that I cannot switch from CD to radio on my friend's car stereo unless I stop the car and grab the manual in the glove compartment, that it is too late to set my digital egg timer when the water is already boiling as it takes too long to adjust it correctly, etc. These problems are perhaps not that serious – they do not harm me physically, at most they annoy and frustrate me and they waste my time in my everyday life. In work settings usability problems tend to be more serious in terms of ineffective use of systems as well as lost data and lost working hours because of wrong usage of the systems, not to mention extended and costly learning effort to use new types and new versions of systems. Even more fatal are usability problems when they are found in high-risk systems, like in operating rooms in nuclear power plants, in fighter airplane cockpits, air traffic control rooms, etc.

In these places a system, which might work perfectly well with respect to technology and functionality, might lead to catastrophes with hundreds or perhaps thousands of lost lives due to usability problems. There are many such disasters like the Three Mile Island accident in 1979, the Chernobyl accident in 1986 (Reason, 1990), the TWA flight 800 accident in 1996 (Schmid, 1998) and many more alike.

Now we have seen examples of usability problems, but how do we define usability more precisely? A precondition for a successful system is that it has a certain degree of utility and a certain degree of usability. Utility is the extent to which a system does what is needed functionally. Usability is a term with various definitions. Shackel (1991, p. 25) defines usability in terms of effectiveness, learnability, flexibility, and attitude. Jordan (1998) defines usability as guessability, learnability, experienced user performance, system potential (theoretically, how efficient can a given task be completed with the system in question), and re-usability. One definition elevated to a standard is found in ISO/DIS 9241-11, which defines usability to be *Effectiveness*, *Efficiency*, and *Satisfaction*. Effectiveness is the accuracy and completeness with which specified users can achieve specified goals in particular environments. Efficiency is the quantity of resources spent in comparison with the accuracy and completeness of the goals achieved. Satisfaction is the comfort and acceptability of the system to its users and other people affected by its use. In our work we will use the ISO/DIS 9241-11 standard as the definition for usability.

From knowing what usability is, we are, unfortunately, far from knowing how usability might be ensured and improved in systems development. The next section describes three approaches to ensure and improve usability.

1.1.2 How to develop usable systems: Important approaches

Accepting that developing usable systems was difficult, naturally led to several attempts to overcome the problem. Although a few pioneers within computer science foresaw how computers should be constructed in order to be useful and usable e.g., Licklider (1960), important contributions in the field of Human-Computer Interaction (HCI) came from other fields than computer science. The following three approaches had impact on the development of UEMs:

1) Industrial experiences and research experiments were gathered to develop *usability guidelines* that could be used in the development process.
2) From a psychological perspective Card, Moran & Newell advocated a theoretical approach based on the *Model Human Processor* (Card et al., 1983).
3) With a *process perspective* Gould & Lewis (1985) and others conducted case studies in real life settings with the aim of finding ways to construct usable systems.

In this section we will describe these three main approaches to improve the interaction between user and system. With this background knowledge, we will be able to describe how the field of UEMs was inspired by these three approaches. The three approaches do not cover all aspects of HCI, rather they reflect what is important for understanding what UEMs are and to explain why we have chosen to work with these methods.

A straightforward way to improve the usability of systems was to develop usability guidelines. These guidelines comprised general design recommendations, interaction system considerations, display techniques, data input considerations, guidelines for response time, etc. Table 1 shows extracts of human factors guidelines used at NCR (The National Cash Register Company) in the eighties (Keister, 1981). Many leading companies developed their own human factors guidelines made for their specific products (e.g., at IBM, Engel & Granda, 1975). The guidelines were devel-

oped on the basis of experience, observations and rigorous experiments on the interaction between users and system (e.g., Grudin & Barnard, 1984; Kiger, 1984). With funds from the U.S. Government, Smith & Mosier (1986) published the so far most comprehensive collection of human factors guidelines comprising almost a thousand specific guidelines. At first blush, the guidelines had the potential to prevent serious usability problems. However, they were difficult to apply for at least four reasons. First, it was difficult to know which guidelines to use in which situations especially when the designer had to violate one guideline in order to follow another guideline. For example, in Table 1, there is a contrast between the two guidelines "Allow alternative modes of operation" in the second column and "Avoid user errors" in the last column. Alternative modes of operation (e.g., the Caps Lock key on computers) tend to attribute to so-called mode errors (e.g., WHEN cAPS lOCK IS MISTAKENLY ACTIVATED), which is not exactly a way to avoid user errors. Second, the sheer volume of guidelines in reports made them difficult to apply in real-life settings. Third, it is not obvious how a system component should be designed to follow a certain guideline. It is, for example, obvious that data should be easy to understand, but *how* is this ensured? The guidelines seldom give any specific suggestion to this problem; guidelines are general in content, but not helpful when designing a specific interface component. Fourth, guidelines become outdated with time as guidelines are often formulated with respect to the technology available.

Guidelines are still used in industry and taught in Human-Computer Interaction classes, and they are definitely better than having no tools at all. However, the problems with using usability guidelines triggered researchers to develop better methods to ensure usability; methods that tended to structure the usability effort and were more resistant to the type of technology tested.

General design recommendations	Interactive system considerations	Display techniques	Data input considerations
• Consider the user • Evaluate displays and dialogues continually • Distribute control functions properly • Make the job interesting • …	• Allow alternative modes of operation • Maintain consistency • Provide constant feedback • Provide useful on-line information • …	• Only display relevant information • Make data easy to understand • Make messages brief and clear • Make menus easy to use with clearly marked alternatives • …	• Make error messages contain information on how and where the error occurred and how to correct the error • Make change and error correction procedures simple • Use standard methods • Avoid user errors • …

Table 1. Extracts of human factors guidelines used at NCR in 1981 (Keister, 1981). Usability guidelines were an early attempt to ensure usability of products.

Since guidelines were limited regarding impact and usefulness, other researchers aimed at building user models. In the mid- and late seventies Card, Moran & Newell worked on developing models of expert performance based on psychological theory. The Model Human Processor (MHP) was the anchor point of their research. The MHP is an approximate cognitive model of the user to be employed by the designer in thinking about the human-computer interaction. The model permits some simple approximate calculations, such as how fast touch typist can type and how fast experts can correct specific errors in a word processing application. Based on the MHP, Card, Moran & Newell (1980a) developed a family of GOMS models, which characterized user behavior in terms of user *Goals*, *Operations* that the user could perform, *Methods* for achieving the goals, and *Selection* rules for choosing among alternative methods. GOMS is able to model expert performance and, given an interface, making approximations about time to perform a given task. Card, Moran & Newell also developed the Keystroke-Level Model (KLM) (Card, Moran & Newell, 1980b), which is a simplified, practical instance of the GOMS family. Their book covering the basic GOMS family, *The Psy-*

chology of Human-Computer Interaction (Card, Moran & Newell, 1983), was a theoretical breakthrough and a milestone in HCI (see also John & Kieras, 1996). However, their book also received hefty critique. Shneiderman writes for example: "*The KLM applies the reductionist method to its limit, dismissing vital factors, that, I believe, influence user performance*" (cited in Newell & Card, 1985).

Newell & Card (1985) themselves were concerned about their work by pointing out four points of difficulties with their approach.

1) The science is too low level
2) The scope of the science is too limited
3) The science is too late
4) The science is too difficult to apply.

Not too surprisingly, they defend these criticisms by arguing that although there are problems using this approach, it is the most promising in HCI. The essence of their argumentation is that psychology in HCI needs to be hardened like other science fields (e.g., computer science and artificial intelligence) in order to have any real impact in the design process. Thus, they defend the four criticisms by arguing

1) that psychology in the very fact tends to deal with low level issues,
2) that the scope of science will expand with time,
3) that science can in fact keep up with technological innovations if science is driven by proper theory, and
4) that we, with time, will see models running on computers that can be useful in real-life settings.

A year later Carroll & Campbell (1986) represented a completely different viewpoint than what was proposed by Card, Moran & Newell. They argued that the premise, that hard science drives out soft science, is wrong, and that Newell & Card's implication that hard science is good science is even more wrong. Carroll & Campbell agreed that the four criticisms proposed by Newell & Card were correct, while Newell & Card's suggestions to solve these criticisms were flawed. Carroll & Campbell continued making propositions for why Newell & Card's defense attempts were not valid. Hence, not only did Card, Moran & Newell's work produce a number of HCI models, but they also raised a discussion in the HCI community about how HCI research and practice should evolve.

Some researchers found Card, Moran & Newell's work interesting and attempted to develop UEMs (rather than models of performance) that were inspired by the reductionist point of view. One of the two methods that we have investigated, the cognitive walkthrough, is in fact a UEM inspired by the idea that users can be modeled in a way that enable evaluators to predict users' interaction with a given system.

Human Factors guidelines and the Model Human Processor were very specific attempts to ensure the development of usable systems. Both approaches focussed on a model or a tool, rather than the process of developing systems. Gould & Lewis (1985) used the latter approach, focusing not on one specific model, method or tool, but on the complete process of developing systems. They initially proposed three design principles:

1) Early and continual focus on users,
2) Early and continual empirical measurements of usage, and
3) Iterative design.

Gould (1988) added yet another general principle:

4) Integrated design.

The first principle – early and continual focus on users – is important because the designer cannot design a system if he[2] does not know who the user is and what he wants. Methods to carry out early focus on users includes talking to users, observing users, learning about the work organization in which the system is intended to run, employ participatory design, inviting domain experts into the development process, conducting tasks analysis, and collect information via surveys and questionnaires.

The second principle – empirical measurement of usage – is important because "[…] *it is more realistic to ask designers to pause at suitable points in the design process, and then to review and evaluate their present results consciously and systematically, than it will be to try to formalize the design process itself*" (Pejtersen & Rasmussen, 1997, p.1515). Gould (1988) proposed empirical evaluation by showing users initial paper and pencil drafts of the user interface, and possibly videotape their simulated use of the paper prototype, by early demonstration of the system, by hallway and storefront methodologies (showing the initial system in public spaces and ask for comments from bypassing people), by formal prototype testing, through try-to-destroy-it contests, and by conducting field studies and follow-up studies.

The third principle – iterative design – is necessary because one cannot do it right the first time. Heckel (1984) asks "*If Ernest Hemingway, James Michener, Neil Simon, Frank Lloyd Wright, and Pablo Picasso could not get it right the first time, what makes you think you will?*". Iterative design can be ensured by using software tools developed for that system engineering approach, but generally iterative design is merely an acceptance of the need for evaluation *and* to take the results of the evaluation seriously. That implies that evaluation is not conducted to accept that there are problems but to re-design those aspects that were proven to be problematic and if necessary abandon the evaluated design solution in favor of a better design suggestion.

Finally, Gould's (1988) last principle – integrated design – was added to the three earlier design principles because case studies revealed that the lack of coordination between various usability initiatives in itself contributed to usability problems. If functions are changed in the system due to usability test results, some coordination has to be done to ensure that this change is reflected in the help system, the reading material, the training material, etc. This principle is of organizational concern, as preferably one person rather than a board of people, should coordinate the usability initiatives in a project.

How do the three approaches described in this section contribute to the development of UEMs? First of all, human factors guidelines were found to be inappropriate and not as usable as one would like a tool to be; practitioners needed better tools to ensure the development of a usable system. Second, the Model Human Processor was the first serious attempt to apply cognitive psychology to Human-Computer Interaction. One of the UEMs that we have worked with, the *cognitive walkthrough*, is a spin-off or at least a method inspired by the formal methods developed by Card, Moran & Newell. Third, Gould & Lewis advocated for a broader perspective in HCI. One of their four principles regarded early and continual empirical evaluation. Their work inspired many researchers to develop specific UEMs that could meet the demand of this principle. In particular did

[2] To avoid favoring any of the two genders and to somehow be consistent in the use of gender, users and systems designers will be denoted "he", while usability evaluation method developers, evaluators, and any other persons will be denoted "she".

Lewis (1982) as the first within the HCI community describe the *usability test*, which is the other UEM that we have been working with. In the next section we will describe the history of UEMs and in chapter 2 describe more closely the two UEMs we have worked with during the past three years.

1.1.3 Focusing on Usability Evaluation Methods (UEMs)

Usability evaluation methods (UEMs) are used to evaluate the interaction of the human with the computer for the purpose of identifying aspects of this interaction so that they can be improved to increase usability (Gray & Salzman, 1998). Three types of UEMs have been identified: *empirical* methods, *inspection* methods, and *inquiry* methods. Empirical methods are based on users' experience with the system in question collected in some systematical way. Inspection methods are conducted by usability specialists – and sometimes software developers, or other professionals – who examine usability-related aspects of a user interface without involving any user. Inquiry methods focus on information about users' likes, dislikes, needs, and understanding of the system by talking to them, observing them using the system in real work, or letting them answer questions verbally or in written form. Table 2 shows various UEMs divided into the three types of UEMs.

Method category	Name of method	Described in (among others)
Empirical methods	Usability test (also called thinking aloud method)	(Lewis, 1982)
	User performance test	(Nielsen, 1993)
	Remote usability test	(Hilbert & Redmiles, 1998)
	Beta test	(Smilowitz et al., 1993)
	Forum test	(Smilowitz et al., 1993)
	Cooperative evaluation	(Wright & Monk, 1991b)
	Coaching method	(Nielsen, 1993)
Inspection method	Expert review	(Hammond et al., 1984)
	Heuristic evaluation	(Nielsen & Molich, 1990)
	Cognitive walkthrough (CW)	(Lewis et al., 1990)
	Pluralistic walkthrough	(Bias, 1994)
	Structured heuristic evaluation	(Kurosu et al., 1999)
	Perspective-based inspection	(Zhang et al., 1998)
Inquiry methods	User satisfaction questionnaire	(Kirakowski & Dillon, 1988)
	Field observation	(Gould, 1988)
	Focus group	(Zirkler & Ballman, 1994)
	Interviews	(Gould, 1988)

Table 2. Different usability evaluation methods (UEMs) divided into three groups: empirical methods, inspection methods, and inquiry methods. A reference in the last column points back to those who have invented or studied that particular UEMs.

The advantage of empirical methods and inquiry methods is that they, by involving users, tend to produce more ecological valid results than when using inspection methods (ecological validity is the extent to which a method setup resembles the environmental situation of the phenomenon under investigation). The disadvantages of using empirical methods and inquiry methods is that they require great competence and experience in order to achieve useful results and they usually cannot be used before at least a prototype is available. Inspection methods are often advocated as low cost methods, which can be applied to an interface early in the development cycle. Compared to empirical and inspection methods, inquiry methods tend to identify broad usability problems or sometimes just opinions about the systems as a whole. Hence, this type of information might be valuable for marketing or sales divisions, but they might be less valuable for systems developers due to lack of specificity for a given problem.

The empirical methods have their roots in psychological experiments. The usability test, for example, is derived from a psychological method used for investigating human problem solving;

Duncker (1945) among others used this early psychological method, called verbal introspection, to investigate human problem solving. Another empirical method is the user performance test that is used to test whether a sample of users is able to learn and use certain functions in a system within a preset time. Hence, it is a test requiring a hypothesis, which is tested and analyzed by use of experimental techniques. The inspection methods were derived from various traditions. The expert review is an unstructured review of a system with no particular requirement apart from requirements to the evaluator, who should be familiar with the field of usability. Heuristic evaluation is slightly more structured than an expert review, as ten general guidelines are used to support the evaluators' evaluation of a system. Pluralistic walkthrough, structure heuristic evaluation and perspective based inspection are all variants of heuristic evaluation. Cognitive walkthrough (CW) is quite unusual, as it is firmly rooted in cognitive theory. Problem solving theories (Newell & Simon, 1972), theories for human exploration (Lewis, 1988), and the cognitive complexity theory (Kieras & Polson, 1985) as well as aspects of GOMS (Card et al., 1983), ACT* (Anderson, 1983), and SOAR (Laird et al, 1987) were the basis for developing a theory called CE+ (Polson & Lewis, 1990). CW was then developed as a method based on CE+. CW is now in its third version, and has thus not only survived for almost 10 years, but also has received much attention in research settings.

When a UEM receives attention in research settings it has survived its first critical phase. However, the aim for UEM developers is to make it usable in real-life settings. Companies do not generally report which UEMs they are using, apart from when they – at rare intervals – publish experience-based reports. Although some of the UEMs mentioned in Table 2 have been used somewhat in the industry, most have not succeeded to be incorporated into system development practices yet. Among the empirical methods, beta test – which does not solely test usability but also utility and other aspects of a system – and usability test are probably the most widely used in the industry (see e.g., Bawa, 1994; Borgholm & Madsen, 1999; Buur & Bagger, 1999; Bærentsen & Slavensky, 1999; Denning et al., 1990; Dillon et al, 1993; Dolan & Dumas, 1999; Dumas, 1989; Jørgensen, 1989, Jørgensen, 1991; Mills, 1987; Muller & Czerwinski, 1999; Salzman & Rivers, 1994; Sullivan, 1996; Vainio-Larsson & Orring, 1990). Among inspection methods, expert review and heuristic evaluation are probably the most widely used methods (Brooks, 1994; Mack & Nielsen, 1993; Nielsen, 1995). None of the inquiry methods are used widely in the industry. In general, it is probably still the case that most development projects do not use UEMs at all. One reason is that many development projects have no tradition of including UEMs into the developing process. Another reason is that several UEMs are not yet ready to be used in the industry. Moreover, developers lack knowledge of UEMs and are in lack of resources in general because of tight deadlines. With increasing pressure from customers demanding better usability of technology, the industry also begins demanding useful UEMs. Meanwhile, UEMs are evaluated and further developed in research settings. Therefore, some of the UEMs available today but not used in the industry will probably be used in the future.

With the many UEMs available, it is valuable to characterize good UEMs from bad UEMs, since developing a UEM does not guarantee that the UEM will work at all. The quality requirements of a UEM are in many respects similar to the usability requirements of a system. It should be effective, efficient, and satisfactory. Effectiveness is the degree to which it will capture problems that it is intended to capture. Efficiency is the quantity of resources spent compared to the quality of the results. Efficiency, hence, is the costs of learning how to use a given UEM and the time necessary to apply it to an interface compared to its effectiveness. Efficiency is a prime issue in real-life settings. Satisfaction is the comfort and acceptability of the UEMs to its users and other people affected by its use. A tedious UEM will have difficulties being used in real-life settings, even if the UEM is effective and efficient. Also, a UEM that is perhaps challenging to use, but has little persuasive

power with respect to developers and marketing people will often be abandoned in favor of better UEMs. Finally a UEM should be both reliable and valid. Reliability is a measure of whether two successive sessions with a given UEM yield the same results, while validity is a measure of the extent to which the results of a given UEM match what was intended to capture using that particular UEM.

The quality requirements to UEMs, however, are conflicting. A reliable and valid UEM will often not be efficient and satisfactory. On the contrary, a UEMs that is efficient – that is easy to learn and use – might not be reliable and valid. It is an ongoing debate as to which is the most important requirement when developing UEMs for the industry. On one extreme some argue that a little usability is better than no usability; this group of people have invented so-called discount usability methods, e.g., heuristic evaluation (Molich & Nielsen, 1990) and cooperative evaluation (Wright & Monk, 1991b). Other researchers and practitioners favor reliability and validity over efficiency insisting on a high degree of robustness when using UEMs (Gray & Salzman, 1998; Lewis, 1994).

Although reliability, validity, effectiveness, efficiency and satisfaction are theoretical quality requirements to UEMs, some practitioners and researchers have asked other questions in their search for separating good UEMs from bad ones. One prevalent type of study compares UEMs to one another. Most often comparative studies on UEMs employ a quantitative approach comparing number of problems detected with different UEMs while holding the systems evaluated constant (e.g., Cuomo & Bowen, 1994; Desurvire, 1994; Hammond et al., 1984; Jeffries et al, 1991; Karat et al., 1992; Nielsen & Philips, 1993). As will be explained in the next section, the inappropriateness in the methods employed when conducting comparative studies on UEMs has been one major motivation for studying the learning and usage of CW as well as studying one reliability issue of both CW and usability test.

1.2 Motivation

The journal *Human-Computer Interaction* recently devoted a special issue to a detailed review of comparative studies of UEMs. The contents of this special issue were Gray & Salzman's (1998) paper "*Damaged Merchandise? A review of experiments that compare usability evaluation methods*" and ten distinguished researchers' comments (Olson & Moran, 1998). Gray & Salzman were the first to publish the claim that we do not know as much about UEMs as we thought we did. Five well-reputed papers describing comparisons of different UEMs (in particular comparisons between usability test, CW, and heuristic evaluation) were reviewed and found methodologically flawed. The comparative studies were not criticized so much for their aim of comparing UEMs, but merely for their inappropriate use of methods employed in investigating their research questions. Several threats to the validity of the five studies were found, in particular statistical conclusion validity, internal validity, construct validity, external validity, and conclusion validity. Only a couple of the ten commentators refused the mere results of Gray & Salzman's review, but there were very different opinions among the commentators on the importance of these results and there were different opinions on suggested agendas for investigating UEMs in the future. Although Gray & Salzman's work was not the only reason why I found it interesting to work with UEMs, their work greatly encouraged me to study UEMs with regard to learnability, usability, and reliability.

A typical setup in comparative studies of UEMs has been to request a few groups or individuals to apply each a different UEMs to the same interface. A simple counting has then revealed that one evaluator using her UEM found more or less usability problems than other evaluators did applying other UEMs to the same interface. This often led to the conclusion that UEM *x* was more effective than UEM *y*. The obvious question to such studies is: Did they measure the evaluators' ability to

detect usability problems or did they – as they intended – measure the UEMs' ability to identify problems? When varying both evaluators and UEMs it is not clear what the researchers have been measuring. This problem motivated us to study the *evaluator effect* on two UEMs, namely CW and usability test.

As noted by Gray & Salzman (1998), the validity problems in comparative studies of UEMs are serious. However, as noted by some of the commentators to Gray & Salzman's paper, there might be more valuable ways of achieving information about UEMs than simply comparing their abilities to detect problems. Even if we could trust the results of comparative studies, the results would still not increase our understanding about *why* they are different, and *how* we might improve UEMs in order to make them more effective, more efficient, or more satisfactory. To answer such questions we need to conduct qualitative studies of how UEMs are learned and how they are used. This kind of information is not achieved by conducting experiments but by preparing case studies and other types of qualitative studies. We have studied the learning process and the first-time usage of CW through case studies in order to understand the potential of this particular UEM.

With the many UEMs available one could ask why we have chosen to study CW and usability test rather than some other UEMs. There are several good reasons for our choice. The usability test is undoubtedly the oldest and best investigated UEM available. In fact, it is often considered as the golden standard within UEMs. The usability test is sometimes compared to other UEMs. However, most often it is used as a yardstick to measure the effectiveness and efficiency of other UEMs (e.g., Bailey et al, 1992; Connell & Hammond, 1999; Cuomo & Bowen, 1994; Desurvire, 1994; Henderson, 1995). Hence, it is implicitly assumed that the usability test is sufficiently valid and reliable to be used as a yardstick for comparing other UEMs. By investigating the effect of different evaluators we test one reliability aspect of the usability test. If our studies reveal a substantial evaluator effect in usability test, this implies that the results of previous comparative studies in UEMs may just as well be an effect of different evaluators rather than a difference between UEMs. Hence, selecting the usability test – the golden standard of UEMs – as the target method in the evaluator effect study was no hard choice.

There were also good reasons to choose CW as the method to investigate with regard to learnability, usability, and predictive power. CW is one of the few UEMs based on cognitive psychology. Partly therefore, it was received well by the HCI community. Most other UEMs are derived from experience, common sense, or simply folklore type of knowledge. Hence, when evaluating and further developing these non-theory-based UEMs there is no stable ground to rely on. An appropriate theory is a more stable ground as foundation for developing UEMs. CW has received much attention in the literature (currently I have collected more than 50 papers dealing with this UEM), but it has not to any large extent found its way into the industry. This is one reason to investigate CW more closely to reveal what might be problematic with the method. Another aspect that makes CW special is that the same group of people has been developing and elaborating on the method for almost ten years. This is indeed unusual in a field that in itself is very young and that tends to have a short "collective memory".

After conducting the two studies mentioned above we felt encouraged to conduct a third study, namely investigating the evaluator effect on CW. CW was advocated as a structured technique with an evaluator simulating a user's usage of a system with the aim of detecting problems related to ease of learning. If we could find an evaluator effect for CW this would have extensive implications for the usage of the method, and would also possibly point in directions for further development and improvement of CW.

In summary, Gould & Lewis (1985) proposed continual evaluation of systems design as one way to improve and ensure usable products. However, the UEMs used to evaluate systems design should be of good quality. We wanted to contribute to a better understanding of CW with respect to learnability and usability through use of case studies, following the tradition of John & Packer (1995) and John & Mashyna (1997). We also wanted to follow the traditions of Barker & Biers (1994), Cantani & Biers (1998), Hackman & Biers (1992), Held & Biers (1992), and Ohnemus & Biers (1993) who studied particular aspects of usability test in experimental settings in order to increase the understanding of the strengths and weaknesses of this method. Our interest has been to reveal the evaluator effect in cognitive walkthrough and usability test.

1.3 Research agenda

The motivation mentioned in the previous sections leads us to our research agenda:

- We wish to investigate qualitatively the learning process and the first-time usage of cognitive walkthrough (CW) in order to achieve a more knowledge about this particular evaluation process and to come up with recommendations for improving the method. The study will be named "*learning and using cognitive walkthrough*"
- We wish to study the evaluator effect in cognitive walkthrough to test this particular reliability aspect of the method and to hypothesize about plausible causes for the results. This study will be named "*the evaluator effect in cognitive walkthrough*"
- We wish to study the evaluator effect in usability test to measure this particular aspect of reliability of the method and to hypothesize about plausible causes for the results. This study will be named "*the evaluator effect in usability test*"

1.4 Contents and reading guide

This thesis consists of this summary and appendices containing a submitted journal paper (Jacobsen & John, 1999), three published papers (Jacobsen et al., 1998a; Jacobsen et al., 1998b; Hertzum & Jacobsen, 1999), and an unpublished manuscript (Hertzum & Jacobsen, 1998). The substance of my work is documented in the appendices. However, this summary attempts to bind my work together to a coherent narrative.

- Chapter 1 (this introduction) ends with this contents and reading guide.
- Chapter 2 is a general description of CW and usability test.
- Chapter 3 and appendix 1 covers the studies on learning and using CW.
- Chapter 4 and appendices 2 and 3 cover our work on the evaluator effect in CW.
- Chapter 5 and appendices 4 and 5 cover our work on the evaluator effect in usability test.
- Chapter 6 is a general discussion on our work.

See Figure 2 for an illustration of the connection between chapters in this summary and appendix numbers in the appendices.

The three published conference papers about the evaluator effects (Jacobsen, Hertzum & John, 1998a; Jacobsen, Hertzum & John, 1998b; Hertzum & Jacobsen, 1999) were restricted with regard to number of pages (2, 5, and 5 pages respectively), while the unpublished manuscript was a brief, initial report (8 pages). Due to these circumstances, I have chosen to elaborate on the reporting on the evaluator effect in CW and the evaluator effect in usability test in this summary. But I have not felt it necessary to elaborate on the contents or reporting of the case study of learning and using CW, since we have not had any page limitations here. Hence, chapter 3 in this summary is a re-

porting of my own work in learning and using CW (Jacobsen, 1996, Jacobsen, 1997), which is also an introduction to reading the comprehensive case study in appendix 1 submitted to the journal *Human-Computer Interaction*. With this structure, this summary is solely my work, while the appendices contain reports written in cooperation with other researchers.

The reader is advised to read chapters 1, 2 and 3 in this summary, followed by reading appendix 1 before continuing with the remaining chapters 4, 5, and 6 in this summary. Appendices 2 and 3 are earlier reports on the evaluator effect in CW, the former (Hertzum & Jacobsen, 1998) being more extensive than the latter (Hertzum & Jacobsen, 1999). Appendices 4 and 5 are earlier reports on the evaluator effect in usability test; the former (Jacobsen, Hertzum & John, 1998a) describing results of the initial evaluator effect study, the latter (Jacobsen, Hertzum & John, 1998b) describing results of an extension to this study. These appendices are valuable to skim through in order to determine which results and discussions in the summary chapters are taken from earlier reports and which are new analyses unique to this summary. For the reader with prior knowledge in UEMs, who seek a quick way to acquire the essence of the evaluator effect studies, appendices 3 and 4 would be sufficient reading as a replacement for chapters 4 and 5 in this summary. Thus, chapters 4 and 5 can be understood without reading the underlying appendices, just like appendices 2-5 can be read without loosing the major essence of chapters 4 and 5. These chapters contain the aspects of the evaluator effect studies that *I* regard as the most important. Hence, a few aspects discussed in the appendices have been left out of chapters 4 and 5, while a few new aspects have been added; aspects which I have analyzed after the appendices were published.

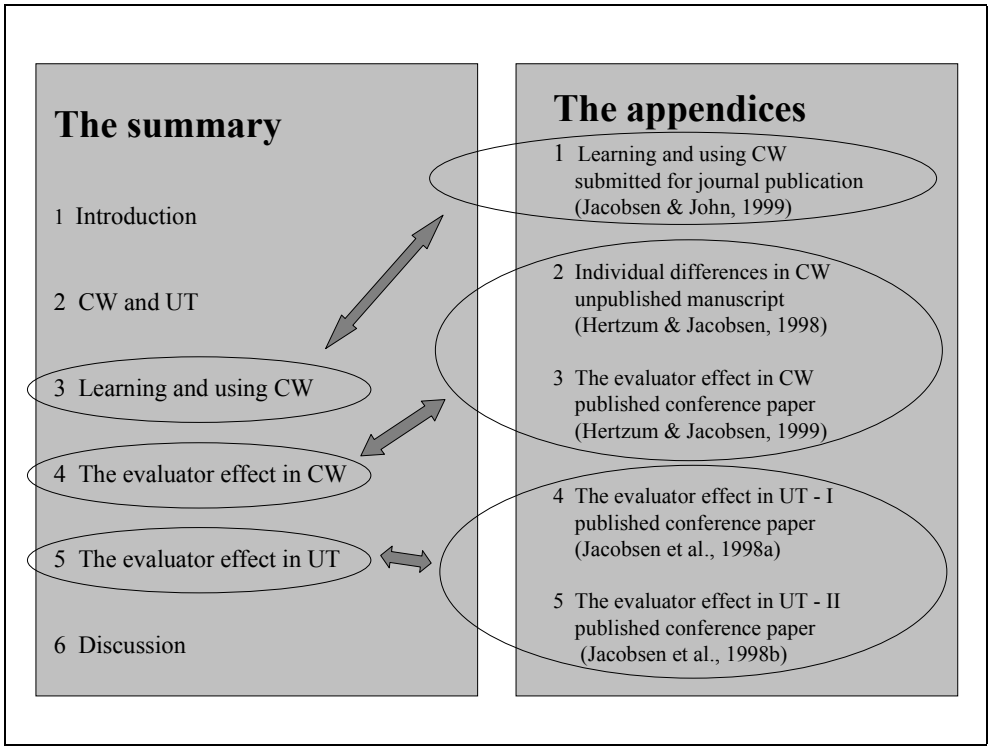

Figure 2. The contents of this thesis and the connections between chapters in the summary and appendices. The reader is advised to read chapter 1, 2, and 3 in the summary followed by the case study paper in appendix 1 before reading the remaining chapters 4, 5, and 6 in the summary. The contents of the remaining appendices 2, 3, 4, and 5 are covered in summary chapters 4 and 5. (CW = cognitive walkthrough; UT = usability test).

2 The Cognitive Walkthrough and Usability Test

The aim of this chapter is to present the two UEMs that we have investigated: the cognitive walkthrough (CW) and the usability test. The two sections dealing with the two methods are organized differently since CW is still being developed and has not been used widely in the industry; the usability test is more mature and has been transferred to the industry with reasonable success. Hence, the section on CW focus on the theory behind the technique, the evolution of the technique, and some highlights on earlier research on CW. The section about the usability test is primarily focused on how the method is described to be used.

2.1 The Cognitive Walkthrough (CW)

As outlined in Table 2 (p. 7), CW is an analytic UEM. Some analytic UEMs have been developed with an assumption that the evaluators possess a common-sense knowledge of systems design and HCI (e.g., when using heuristic evaluation) or some particular tacit knowledge that permits them to act as experts in the evaluation process (e.g., when using expert review). CW has not been developed with such an assumption as CW is specifically built on top of a cognitive model or theory about a user's learning process when operating a new system. In the next three sub-sections we will describe the theory underpinning CW, the evolution of the method, and describe how other researchers have investigated CW.

2.1.1 The theory underpinning CW

Some of the first cognitive models appearing in the literature were theory-based and focussed in particular on expert performance. GOMS (an acronym for Goals, Operations, Methods, and Selection), for example, aims to model an *expert's* error-free performance for a given task on a given system (Card et al., 1983). On the contrary, the aim of developing CW was to invent a theory-based UEM that specifically focussed on *novices'* explorative learning of computer interfaces. Another aim for developing CW was to come up with a UEM that could be used with much less effort than traditional cognitive modeling techniques like GOMS, SOAR (Laird et al., 1987) and ACT* (Anderson, 1983).

Chronologically, researchers first conducted basic studies examining human behavior in various situations. They then continued by developing specific theories, e.g., problem solving theories (Newell & Simon, 1972) and models such as GOMS (Card et al., 1983), Cognitive Complexity Theory (CCT; Kieras & Polson, 1985), and EXPL (Lewis et al., 1988). Heavily inspired by these theories, Polson & Lewis (1990) invented a model of learning by exploration, the so-called CE+. This model was named CE+ because it used ideas from CCT and EXPL, while the plus signaled that other significant theories had contributed to the model. Based on CE+, Polson & Lewis (1990) reported eight usability principles, which again was the basis for developing CW. The aim here is not to describe all theories and models that contributed to CE+. This work is already clearly stated in Polson & Lewis (1990). Rather, we wish to describe CE+ in an operational manner, outline the resulting usability principles, and briefly discuss the trade-off of this model.

Novice users, with no initial knowledge of the system they are about to operate, have been found to use hill-climbing problem solving strategies, or more specifically, label-following heuristics (Polson & Lewis, 1990). That is, if a user action that is available in the system matches the goal of the user, this action will most likely be executed. The *problem solving component* in CE+ is responsible for choosing actions by using label-following heuristics. When a user has operated some device, whether this was done correctly or not, the system feedback might teach the user something about

how to use an operation or where to find a function. The *learning component* is the second crucial part of CE+. The user, however, does not use the system unless he is motivated to operate the system and combine knowledge of actions with system feedback. Hence, the model needs a third and crucial *execution component* capable of executing rules and coordinating execution of rules with the problem solving component. Below we will describe the three components in CE+ in more detail.

The general idea of the problem solving component is that, faced with a choice of untried actions, CE+ chooses that action whose description overlaps most with its goal. The exact selection can be specified using iterative pseudo code as seen in Figure 3. The first command walks through all immediately available actions in a user interface and remembers those that match the user's goal with a certain threshold. The second command sorts the collected actions in the user interface in order to prepare a final selection of the most prevalent action in relation to the user's goal. The third command selects the top-priority action (the one that best fits the user's goal) given that this action is labeled close to or identical to the user's goal. The last two commands are backup in case there are no obvious actions that matches the user's goal. Here, a random untried action is selected as the one that the user will choose.

```
For all immediately available actions with a description overlapping the goal with more than a given
threshold name those actions i1-in;

Sort actions i1-in such that the action with the most overlapping description to the goal is assigned
to j1, the next most j2 and so forth applying the following rule: untried actions have relatively
lower priority than actions that have been tried before;

If j1 is above a given threshold j1 is the candidate action;

If j1 is not above a given threshold and there exists an untried action then pick any one untried
random action as the candidate;

If j1 is not above a given threshold and there are no untried actions then

  if n > 2 then pick any random jx as the candidate

  otherwise j1 is the candidate;
```

Figure 3. A piece of pseudo program code that finds a candidate action that most closely matches the goal. (Constructed based on text describing CE+ in Lewis et al., 1990)

Imagine the following example of using the problem solving component. A user wants to print out a page in a computer system. First, the user searches the interface for some action that matches the idea of printing out. If a printer icon or a label with the term "print" is immediately available, the user will most likely execute this. However, are there no such actions available, the user begins executing some action that does not immediately lead to a printer function, but that might hide a printer function – e.g., the action "Files". In this search untried actions are selected before actions already known to the user. If no actions seems to accomplish the goal of printing, some random action is tried out in hope of finding the print function.

The problem solving component is described only vaguely in respect to an appropriate threshold and how to judge the similarity of a goal and a description of an action. In this sense there has not been made any attempt to model CE+ on a computer; the model needs a human to judge appropriate thresholds and make the linguistic analysis or mapping between icons on buttons and goal structure. However, components of CE+ in Polson & Lewis (1990) are described in the way computer scientists and engineers typically describe algorithms meant to be run on computers. With sufficient programming effort the CE+ would be able to run on a computer just like SOAR and ACT* can simulate rather complex problem solving situations.

The learning component in CE+ stores information on the value and appropriateness of system feedback each time an action has been executed. The following rule applies: if the system feedback on a given action shares terms with the goal representation, the action is deemed to have been successful. If there is no relation between system feedback and the initial goal representation, CE+ stores the system feedback for future references and automatically undoes the action to make a new search for an appropriate action. To resemble a user's cognitive capacity the storage structure in CE+ decays with time, i.e., the most recently used actions and their feedback are more easily remembered than actions that were executed a long time ago. The contents of the learning component is translated to productions in a production system, which is typical for cognitive modeling techniques like SOAR (Laird et al., 1987). Since humans do not solve problems, if they have solved the particular problem previously, the problem solving component in CE+ is only used when a search in the production system (the memory of the learning component) results in no hits. Hence, problem solving is only something that CE+ invokes when forced to, due to lack of information in the internal storage.

Though goal structures and goal management is not touched upon in the description of CE+ in Polson & Lewis (1990), they implicitly indicate that CE+ inherits the goal structure known from GOMS. That is, CE+ initially seeks to solve a problem. That problem can be broken down into goals that might again be divided into sub-goals. Expanding the goal-structure at all branches reveals a multi-level goal-tree with the problem on top and the lowest level sub-goals as leaves. At the leaf level there will only be a subtle distinction between a goal and an action. In GOMS, an action is not even the lowest level, as actions can be divided into mental preparation units, perception units, physical movements (of the hand for example) and simple reaction time. In CE+, the goal structure is not that detailed as the model operates on goal structures that do not get more detailed than clicking a button or entering in a word.

CE+ was the basis for defining eight usability principles related to designing computer interfaces for novice users. The eight principles are outlined as follows (Polson & Lewis (1990, p. 214).

1. Make the repertoire of available actions salient
2. Use identity cues between actions and user goals as much as possible
3. Use identity cues between system responses and user goals as much as possible
4. Provide an obvious way to undo actions
5. Make available actions easy to discriminate
6. Offer few alternatives
7. Tolerate at most one hard-to-understand action in a repertoire
8. Require as few choices as possible.

The description of CE+ is too abstract to be run on a computer partly because of the lack of absolute thresholds, e.g., a threshold of when a label text matches the goal of a user. This property makes it less attractive than GOMS and CCT regarding precision and predictability. On the other hand CE+ has its strengths because it is informal and reasonably easy to adapt to real situations.

CW was invented to make CE+ more operational, to hide some of the theory underpinning the method, and to enable evaluators with little training and knowledge in cognitive modeling techniques to actually use CW in real-life settings. There is an on-going discussion in the HCI community whether cognitive theory has anything to offer to HCI practitioners (see e.g., Barnard, 1991; Landauer, 1987; Norman, 1995). More specifically, for those who believe that cognitive theory has the potential to be the basis for new achievements in HCI, there are disagreements about *how* it should be transferred to the HCI community. May & Barnard (1995) do not believe that cognitive theory can be wrapped into a technique and used by laymen. They specifically used the case of CW to argue that it has not been successful in transferring cognitive theory into a practical situation.

Obviously, the researchers behind CW made the assumption that CE+ was appropriate as a basis for developing a UEM, and that it would be useful to hide some of the theory behind CW from the users of the technique.

In summary, CE+ is a conglomerate of explorative learning theories of which CCT, EXPL, and a theory of problem solving are the most prominent aspects. The model is vaguely defined giving room for modelers to adjust and use the model in various situations. On the other hand it might produce less precise results than more advanced models like GOMS and CCT. To use the model as it is, one needs good skills in cognitive theory. CE+ consists of three components, a problem solving component, a learning component, and an execution component. The model was the basis from which the method developers reported eight usability principles. The model and the principles are the basis for CW.

2.1.2 The evolution of CW

The first version of CW appeared in a proceeding paper presented at the ACM CHI'90 conference (Lewis et al., 1990). This version of CW was further elaborated and presented in a second version in *International Journal of Man-Machine Studies* (Polson et al., 1992). Two years later the third version of CW appeared first as a technical report (Wharton et al., 1993) and later as a chapter in Nielsen & Mack's (1994) *Usability Inspection Methods*; this third version was called *The Cognitive Walkthrough Method: A Practitioner's Guide* (Wharton et al., 1994). Finally two of the method developers (Lewis & Wharton, 1997) described CW in a chapter of Helander et al.'s (1997) *Handbook of Human-Computer Interaction*. This latest description of CW, simply called *Cognitive Walkthrough*, was presented in a version very much similar to the one in the *Practitioner's Guide*, though the latest paper focussed especially on describing those parts of the technique that had been critisized by other researchers as well as advising practitioners on issues that were obscure in the version of the *Practitioner's Guide*.

In this section, we describe the different versions of CW in order to reveal how the model behind CW, CE+, and the practical usage of CW have driven the CW developers to elaborate and iterate on CW. Describing earlier versions of CW aims at increasing the readers understanding of the conclusions that we draw both from the case studies and from the evaluator effect study. However, we will focus especially on the third version of CW described in the *Practitioner's Guide*, as our studies were based on this version. The latest description of CW (Lewis & Wharton, 1997) is not a new version per se, and hence, in general, the results from our study still apply to the usage of the CW as it is described today.

General Procedural Description of CW

Although CW has changed somewhat since its introduction the general procedure is similar for all versions of the method. First the evaluator or evaluator team specifies the tasks that should be evaluated. Then, the evaluator specifies the correct action sequence(s) to accomplish the selected tasks and the fictive user is defined. Finally, the evaluator walks through the action sequences guided by some questions that have to be asked for each action in the action sequence. If some answers to the questions in the walkthrough procedure seem problematic, a problem has been detected.

The First Version of CW

The general procedural description of CW (described above) was used in the first version of the CW (Lewis et al., 1990). The question form, that guides the evaluator through each action in an action sequence, is presented in Figure 4. The evaluator begins by describing the user's current goal

(question 1) and correct action (question 2). The series of questions from question 2a to question 7 evaluates the ease with which the user correctly selects and executes the action. Then the evaluator describes the system feedback for the action and judges whether it is adequate. Finally question 9 addresses to what extent the user will form an appropriate goal for the next action in the action sequence. Thus, the first CW description comprises judgment about the user's goal and 14 questions[3] that should be answered for each action in an action sequence.

CE+ Design Walkthrough:_____ **Evaluator:**_____ **Date:**_____

Interface:_____ **Task:**_____ **Step #:**_____

Actions/Choices should be ranked according to what percentage of potential users are expected to have problems:
0 = none; 1 = some; 2 = more than half; 3 = most.

1. Description of user's immediate *goal*:

2. (First/next) atomic *action* user should take:
   2a. Obvious that action is *available* to *goal*? Why/Why not?
   2b. Obvious that action is *appropriate to goal*? Why/why not?

3. How will user access *description* of action?
   3a. Problem accessing Why/Why not?

4. How will user *associate* description with action?
   4a. Problem associating? Why/Why not?

5. All other available actions *less appropriate*? For each why/why not?

6. How will user *execute* the action*?*
   6a. Problems? Why/why not?

7. If *timeouts*, time for user to decide before timeout? Why/why not?

8. Execute the action. Describe *system response:*
   8a. Obvious *progress* has been made toward goal? Why/why not?
   8b. User can access needed *info.* in *system response*? Why/why not?

9. Describe appropriate *modified goal*, if any:
   9a. Obvious that *goal should change*? Why/why not?
   9b. If task *completed*, is it obvious? Why/why not?

Figure 4. An exact copy of the first version of the question form (Lewis et al., 1990, p. 237), which is intended to guide CW evaluators walking through each action in an action sequence.

The first version of the question form was later evaluated by the CW developers and others, who found the form to be imprecise and to little help even for evaluators with extensive knowledge in cognitive psychology. They therefore changed CW to a second version much more precise and with much more details in how to walk through action sequences.

The Second Version of CW

The second version of CW was presented in Polson et al. (1992). This paper is much more concise and detailed in the description of both the CE+ model and the CW itself. Here it is specifically claimed that an evaluator without extensive training in cognitive psychology should be able to apply CW to an interface. The CW process is split into a preparation and an execution phase. The preparation phase comprises a selection of tasks, setting up the correct action sequences for accomplishing the tasks, describing the anticipated class of users, describing the initial state of the system, and

[3] We have simply counted the question marks in Figure 4 to find the number of questions in the first version.

defining the user's initial goals. In the execution phase the evaluator analyzes each action in the action sequence in depth (see Figure 5).

Task:_____ **Action#**_____

1. Goal structure for this step

1.1 **Correct goals.** What are the appropriate goals for this point in the interaction?

1.2 **Mismatch with likely goals.** What percentage of users will not have these goals, based on the analysis at the end of the previous step?

2. Choosing and executing the action. Correct action at this step:_____

2.1 **Availability.** Is it obvious that the correct action is a possible choice here?

2.2 **Label.** What label or description is associated with the correct action?

2.3 **Link of label to action.** If there is a label or description associated with the correct action, is it obvious, and is it clearly linked with this action?

2.4 **Link of label to goal.** If there is a label or description associated with the correct action, is it obviously connected with one of the current goals for this step?

2.5 **No label.** If there is no label associated with the correct action, how will users relate this action to a current goal?

2.6 **Wrong choices.** Are there other actions that might seem appropriate to some current goal?

2.7 **Time-out.** If there is a time-out in the interface at this step does it allow time for the user to select the appropriate action?

2.8 **Hard to do.** Is there anything physically tricky about executing the action?

3. Modification of goal structure. If the correct action has been taken, what is the system response?

3.1 **Quit or backup.** Will users see that they have made progress towards some current goal?

3.2 **Accomplish goals.** List all current goals that have been accomplished. Is it obvious that they have been accomplished?

3.3 **Incomplete goals that look accomplished.** Are there any current goals that have not been accomplished, but might appear to have been based on the system response?

3.4 **"And-then" structures.** Is there any "and-then" structure, and does one of its subgoals appear to be complete?

3.5 **New goals in response to prompts.** Does the system response contain a prompt or cue that suggests any new goal or goals?

3.6 **Other new goals.** Are there any new goals that users will form given their current goals, the state of the interface, and their background knowledge?

Figure 5. A copy of the second version of the question form (Polson et al., 1992, pp. 752-755), which is intended to guide CW evaluators walking through each action in an action sequence. The question form presented here is shortened compared to the original question form in Polson et al. (1992). However, only questions quantifying some of the points above have been left out here (in particular a repeated question about how many users would experience a given problem).

Rather than dividing the execution phase into nine parts with a total of 14 questions, as was the case in the first version of CW, the second version divides the execution phase into 3 parts with a total of 17 questions[4]. Only 19 headings and the 17 associated main questions are shown in the shortened version in Figure 5. The questions not shown in the figure ask the evaluators to qualify their answers by judging how many users will experience a problem; these have not been counted as additional questions here.

In summary, the second version of CW is similar to the first version with respect to the general procedure, but different with respect to how many and which questions to answer for each action in an action sequence. The second version is more elaborated than the first version, but because of this extension the CW process became slightly more time-consuming, as the evaluator were requested to answer 17 rather than 14 questions for each action in an action sequence.

[4] We have simply counted the question marks in Figure 5 to find the number of questions in the second version.

This change had consequences for the usability of CW. Consider a simple system, like an ATM machine. A typical task to evaluate would be to withdraw a given amount of money, say 100 US$. A possible action sequence for this task could be:

1. Put in credit card in appropriate slot
2. Enter in pin-code
3. Push appropriate button for accepting pin-code
4. Enter in amount to be withdrawn (100$)
5. Push appropriate button for accepting amount
6. Accept or refuse to have a receipt printed out
7. Remove credit card from appropriate slot
8. Possibly remove receipt
9. Remove money

Using the second version of CW the evaluator were guided to answer 17 questions for each action in the action sequence yielding 153 answers. Using the first version of CW would only require answers to 126 questions. As will be presented below the third version of CW comprised only four rather than 17 questions, probably because the process became too time-consuming to be used in real-life settings; the example above could be walked through with only 36 answers using the third version of CW.

The Current Version of CW

The third version of CW (Wharton et al., 1994) was presented in the book *Usability Inspection Methods* by Nielsen & Molich (1994). In this description of CW the authors motivate the use of CW by the fact that users tend to learn software by exploration, rather than through formal training. CW is not only appropriate for simulating novice users on walk-up-and-use systems, but for simulating any novice user who is exploring a system. The target evaluators are practicing software developers. A CW analysis again consists of a preparation and an execution phase. The preparation phase is similar to the one presented in previous versions of CW. Based on a description of a system or a running system the evaluator should identify users, construct tasks to be evaluated and define the correct action sequences. An example of a fictive user description could be “User is familiar with the Macintosh interface”. The execution phase is done by individuals or in groups. Here, the evaluator should answer only four questions for each action in an action sequence:

1. Will the user try to achieve the right effect?
2. Will the user notice that the correct action is available?
3. Will the user associate the correct action sequence with the effect that the user is trying to achieve?
4. If the correct action is performed, will the user see that progress is being made toward solution of the task?

If anyone of the four questions leads to a negative answer a problem has been detected. If all questions are answered positively, there is not detected any problems for that action and the next action in the action sequence is then analyzed. After each analyzed action the evaluator might record user knowledge requirements, revised assumption about the user population, and notes about design changes. Finally, the problems identified lead to fixes in the software.

2.1.3 Research on CW

CW is one of the few UEMs that has survived and been elaborated on through several versions. It is now described in many HCI textbooks (see e.g., Dix et al., 1998; Jordan, 1998; Newman & Lamming, 1995; Preece, 1994; Shneiderman, 1998) and it is included in one way or another in at least 50 conference proceeding papers and journal papers in the HCI literature. The model underpinning CW (CE+) and its parent theories have been scrutinized (e.g., Knowles, 1988; May & Barnard, 1995), the outcome of using CW has been compared to other UEMs (e.g., Cuomo & Bowen, 1994;

Dutt et al, 1994; Jeffries et al., 1991; Karat, 1994), and practical experiences using CW have been reported (e.g., Brooks, 1994; Ereback & Höög, 1994; Mack & Montaniz, 1994; Olson & Moran, 1996; Rowley & Rhoades, 1992). In this section we will summarize what we think is the most important results from studies conducted on CW. These include criticism on the tediousness of using CW, the method's limited scope, the requirement that evaluators need expertise in cognitive theory, the limited number of problems identified when using CW, and the lack of finding severe problems.

The first two versions of CW were found to be tedious to use due to the many questions that had to be answered for each action in an action sequence. Although CW can be applied early on in the development cycle and only with a specification document at hand (opposed to a running system) it feels like an extremely detailed process for an evaluator to spend perhaps 15 minutes analyzing whether a click on a button would be a problem or not (Jeffries et al., 1991, Karat, 1994). The aim of changing CW and introducing the third version was precisely to cushion this negative effect (however, see Jacobsen (1996) for how CW was still thought to be tedious to use). The reports of tediousness of using CW are massive and exist in almost any study investigating the method, but only few of these were based on the latest version.

Another criticism of using CW is its limited scope. First of all CW focus entirely on ease of learning and although ease of learning is somewhat correlated with ease of use, researchers claim that CW misses other important problems like consistency problems and general problems as well as recurring problems (Jeffries et al, 1991; Wharton et al., 1992). Here, a general problem is a problem that might apply to several aspects of the interface, while a recurrent problem is one that does not only bother the user the first time but keeps being a problem even after the user has some experience with the system. Second, CW is limited to only investigating the correct action sequence for solving a task. That is, CW is claimed to identify a problem if the user diverts from the correct action sequence. However, CW does not identify problems occurring *after* the user has diverted from the correct action sequence, neither does it identify recovery-from-error problems.

A third problem with CW is that the earlier versions of the method are not sufficiently detached from the theory to be used by evaluators with little knowledge in cognitive psychology. This was accepted to be a problem for the method developers at least for the first two versions of CW (Wharton et al., 1994), but in the description of the third version they explicitly claim that CW now can be used by evaluators with little theoretical knowledge. This claim is still to be investigated.

In comparing different UEMs to each other, researchers often used the strategy of counting problems and severe problems to find the most effective UEM. Gray & Salzman (1998) questioned the value of such studies and furthermore found severe methodological flaws in some of the most cited ones (i.e., Desurvire et al, 1992; Jeffries et al., 1991; Karat et al, 1992; Nielsen, 1992; Nielsen & Philips, 1993). Before the publication of *Damaged Merchandise* (Gray & Salzman, 1998), it was believed that CW performed poorly in terms of number of problems identified and the severity of the problems. After Gray & Salzman's paper it seems that we do not know much about CW's performance compared to other methods. Moreover, we do not think that the number of problems is an essential parameter for estimating the value of UEMs, and as described in our studies (see later sections) problem severity is indeed interesting to point out in practice but might be very difficult to identify in a reliable way.

2.2 The Usability Test

The aim of this section is to describe how usability test should be used in a practical work situation. Lewis (1982) was the first to describe the use of usability test in HCI context. Since then researchers have studied many aspects of the usability test and suggested numerous changes to the method

some of which are not commonly accepted. Today there is no definitive definition of the aim and usage of the method, and no specifically accepted procedure to follow, since usability test is used in various situations, with various goals, and both early and late in the development cycle (although see Blanchard (1998) for an attempt to standardize usability tests). Perhaps therefore, it appears in the literature under various names, e.g., "thinking-aloud method", "usability study", "user test", "user testing", and "usability test". To be consistent in this work, I will use the term "usability test" as a blanket term for all versions of the method.

2.2.1 The usability test situation

The usability test situation consists of a user thinking out loud while solving tasks using the system being tested. An evaluator analyzes the user's work and describes usability problems either on-the-fly or through analysis of a videotape. A facilitator arranges the usability test and manages technical aspects of the study as well as communication with the user when necessary. When a number of usability test sessions have been completed, the evaluator reports the results of the study in a usability report and communicates these results to developers.

These steps in a usability test may seem simple and easy to do. However, there have been many questions raised in relation to how a usability test should be conducted. For example: When in the development cycle should usability test be used? Which types of systems can be evaluated? Who should describe the tasks and how should they be described? Which users should participate, how many users should participate, and what kind of background should they have? How should the user think aloud, and how valid are those kinds of data? Which requirements are needed to act as an evaluator or a facilitator? In which environment should the usability test be conducted? What procedure should be followed? How should data be collected and analyzed? And finally how should the results be presented? Understanding the results of our study of the evaluator effect in usability test is dependent on answers to these questions. Therefore, we will try to answer them in the following sections.

Figure 6 shows a dedicated usability laboratory with all important actors and equipment used in a usability test. As will be explained in later sub-sections, a usability test does not necessarily have to be conducted in such advanced settings.

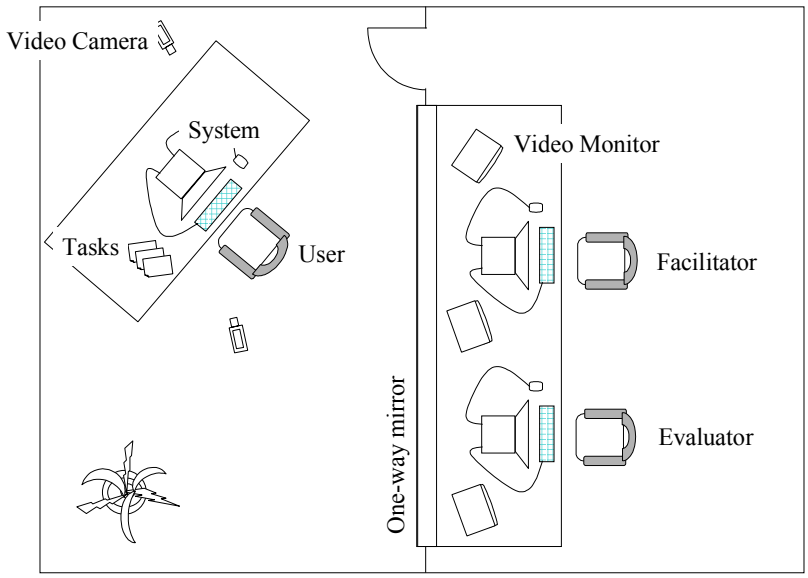

Figure 6. An example of an advanced usability test situation showing a dedicated usability laboratory from a bird's eye view. The user is seated in the chair on the left hand of the figure typically in front of a computer. Besides him are the tasks that he is requested to solve using the computer. The user is videotaped so that the facilitator and the evaluator (on the right hand side) can monitor the user's behavior through their video monitors. Besides this, they watch the session through a one-way mirror enabling them to see into the test room, while the user is relieved from being disturbed by the people monitoring him.

2.2.2 Aim of using the usability test

The aim of using usability test is to identify usability problems to feed into an iterative development cycle in order to correct specific problems or redesign the interface of the artifact in question. Usability tests can also be a vehicle for improving the co-operation between users and systems developers (Buur & Bagger, 1999) to teach systems developers about usability (Nielsen, 1993), or even to improve the PR of the company using the methods (Salzman & Rivers, 1994; Jordan & Thomas, 1994). Although we will not focus on the development process surrounding a usability test, it should be emphasized that a usability test is only one aspect of a complete development process.

In practice there is a close relation between the outcome of using usability tests and when in the development cycle a usability test is conducted. The earlier a usability problem is identified the earlier it can be repaired and the more impact it has on the design of the system (Müller & Czerwinski, 1999). On the other hand, the later in the process a system is evaluated the more the specific problems can be identified with precise suggestions on what to change. Thus, different types of usability tests with different aims should be conducted throughout the iterative development cycle. Rubin (1994) has identified four types of usability tests: explorative, assessment, validation, and comparison.

Explorative usability tests are conducted early in the development cycle with the objective to examine the effectiveness and application of high-level and preliminary design concepts. In explorative usability tests, evaluators might identify problems of a general kind, for example that users have difficulties working with a new type of intelligent windows. Thus, such a result of an explorative usability test might lead to a decision that the concept of intelligent windows needs to be redesigned to something where users have more control.

Assessment usability tests are conducted later in the process than explorative usability tests, but not so late that the problems identified are impossible to fix. Here, the prototype is more advanced and hence, the study focuses more on details in the interface rather than conceptual problems. The aim of using assessment usability test is to expand the findings of the explorative usability test by identifying specific problems. A myriad of problems might be identified in an assessment usability test, for example concerning uniformity of dialog boxes, labeling of menus, procedural problems solving some real-life tasks, missing but needed functions, lack of following standards, misguiding help and documentation, etc.

Validation usability tests are conducted later in the development cycle than explorative studies. Actually the study is often conducted so late that there is little time to repair or fix identified problems. As the name indicates the validation usability test aims at validating some predefined criteria – often quantitatively. From the requirement specification document there might already be identified some requirements, for example, maximum time to perform a given task on the system, maximum time for learning to use a certain function, or some minimum user satisfaction rate. Moreover, the validation usability test might set standards for performance of products that are to be developed in the future. The results from validation tests can be used to check if a contract between developers and customers have been kept or to test if the product has the quality that the company aimed at in the requirement stages of the development cycle.

Comparative usability tests have become more popular recently (see e.g., Dolan & Dumas, 1999). As usability becomes a sales and marketing parameter, sale and marketing staff require that usability staff perform tests that compare competing products. Comparative usability tests are conducted after system release, and hence after a point where problems can be corrected (although they often have impact on the development of the successor to the tested system). The comparative usability test results are used to document the product's quality in terms of usability in comparison with other products.

In relation to studying the evaluator effect we have emphasized the primary aim of using usability tests, i.e., to identify usability problems in the artifact under evaluation. Moreover, we have focussed on explorative and assessment usability tests because these are the most widely used types of usability test in both industry and research.

2.2.3 The artifact tested

In principle any artifact that has a user interface, can be tested using usability test. One popular book on design and usability (Norman, 1988) has several splendid examples of simple everyday artifacts having a user interface, e.g., pocket calculators, clock radios, water faucets, and even door handles. The use of technical books, for example manuals, is also tested using usability test. More typical, however, are usability tests of the usage of software systems running on computers. Theoretically, there seems to be no limits of the use of usability test regarding the complexity of the artifact tested. In reality, however, one cannot expect to test all aspects of a complex and huge system through a usability test. Therefore, the introduction of task scenarios is a practical method for limiting the evaluation of more complex systems (artifacts with little functionality, say a door handle, can be tested without selecting task scenarios).

2.2.4 Task selection

Task scenarios can have various *forms*, be selected for various *reasons*, and selected by different *stakeholders*. The form of a task scenario can be very concrete or rather abstract. If one wants to test the usability of drawing functions in a computer drawing application, one example of a concrete

task scenario is “use function ‘draw line’ to draw a line from upper left corner to lower right corner of the canvas”. A corresponding abstract task scenario could be, “Draw something on the canvas”. Generally, more concrete task scenarios make it possible to compare task completion time and test specific functions; also, concrete task scenarios make it easier for those who monitor the user in respect to following the user’s plans and actions. Meanwhile, concrete task scenarios decrease the ecological validity of the study, as guiding users through an interface via specific tasks might evaluate the usage of the specific functions but without evaluating if the functions could be found in the first place without pointing directly to them via concrete task scenarios. Abstract task scenarios make it increasingly difficult to compare, for example, task completion time, and it increases the variance in what is tested in the system and makes it more difficult for the evaluators to follow the user’s plan of action. However, it also increases the ecological validity as abstract task scenarios to a higher degree enable users to use the system as they would do outside experimental settings.

Task scenarios can be selected for various reasons. The optimal task scenario covers all functions in the system tested. However, in complex systems, task scenarios have to be selected in order to ensure that the usability test sessions can be completed within a reasonable time. Often, tasks are selected to test the use of specific but presumably problematic parts of the system or to test functions often used. One might suspect parts of the system to be problematic for various reasons. Time-critical tasks, interface aspects changed from previous versions of an application, or part of innovative design solutions that need to be tested with real users are good reasons to include these aspects into task scenarios. Also, functions often used should be quick to use with low error rates. Task selection has shown to be a central issue when using any UEM. If less interesting functions are tested and the problematic functions are not covered in the usability test, the effort might be completely wasted. Though task selection is truly critical for the outcome of the usability test there is no final or simple way to ensure appropriate task selection. In particular, testing innovative systems can be really difficult, as no one can predict exactly who would use the system, which functions they would use more often than others, and how they would use the system. In such cases task selection is an art more than a rational choice.

From the analysis phase of the development cycle there might already be task analyses and work analyses available to the usability team. However, when such information is not available, different stakeholders might select task scenarios (e.g., users themselves, facilitators, evaluators, developers, marketing people, or purchasers of the system). If the system tested is a children’s game to be run on a handheld computer, the usability of the game might be tested by letting children themselves (i.e., the users) set up task scenarios. In reality, the users will have a free task scenario like, “please, play a game on this machine”. Game computers are to be used by children without any instructions, and if some children have problems playing a game or using specific functions in the game, a usability problem has been detected. In some organizations, usability staff might have the necessary knowledge to set up task scenarios or they achieve appropriate knowledge through interviews and observations in real-life settings. Part of an evaluator’s job is to run usability tests, which might include setting up task scenarios. Through experience they understand the importance of formulating good task scenarios, and they refine their skills in this respect. However, there is a risk that they lack sufficient knowledge to select the most interesting task scenarios, especially if the domain is unknown or is too complex to grasp in a few days of preparation. Depending on the degree of participatory design in an organization, the developers might know more about the domain in question than what usability staff can obtain in a few days. Moreover, developers are in possession of extensive knowledge of the system developed. This valuable insight implies that developers have the potential to select task scenarios that cover most of the system and gets around specifically difficult or new functions. However, developers are typically focussed on the implementation of the appli-

cation rather than focussed on the real world of tasks to be solved with the application. Thus, developers setting up appropriate task scenarios in relation to the developed system might miss important aspects of real-life settings (even when participatory design is used).

Marketing people and management might have an interest in selecting task scenarios for usability tests if results from these studies are important in relation to sales of a product, typically in competitive usability tests or validation usability tests. If a usability test can prove that, say one expensive library system is easier and faster to use than a competing but cheaper library system, the costs of teaching the librarians using the cheaper system might well exceed the costs of buying the expensive system. Thus, marketing and management are important stakeholders in validation and comparative usability tests, and might be the right ones to select task scenarios.

As described above, any participant in a usability test has some interest or some good reason to be the one to set up appropriate task scenarios. Meanwhile, these participants are rarely unbiased or have any superior qualifications to set up task scenarios. One approach to ensure setting up appropriate task scenarios is to invite all participants to a focus group meeting (Zirkler & Ballman, 1994) or to workshops (Gardner, 1999). Thereby many perspectives are heard, which limits the risk of setting up task scenarios that only consider a limited number of aspects of the system in question.

2.2.5 User selection and user background

Selecting the right users for a usability test is essential for the validity of study results. In general, participating users should have a profile that fits real users of the product tested. If the application tested is a back office system in a large bank, the participating users should be selected from one or more of the offices where the system is intended to be used. These selected users should not have any special status, like being managers or having computer background different from the average user. In order to aim at ecological valid results, appropriate user selection is essential. In this respect, user selection in a usability test resembles selection of participants in traditional psychological experiments, where internal selection validity is crucial when conclusions based on the experiment are drawn.

Practically, user selection is difficult for at least three reasons. First, if the system tested is intended for a large and diverse user group (e.g., when the system is a new cellular phone) representative user selection is almost impossible. Nobody knows who would actually buy the new product, and there might be so many different user profiles from different countries and even different cultures, that only a minority of these profiles will be represented. Second, real users are sometimes not available for the usability team. Thus, the usability team has to select representatives that either do not fit the actual user profile or is not expected to use the system when it is released. Information systems for top managers are being developed, but getting hold of top managers that wish to spend half a day acting as users in a usability test might be extremely difficult. Third, there might not be resources for conducting usability tests with sufficient number of users to meet the requirements for scientific rigor results. Nielsen (1994), Lewis (1994), Virzi (1990), and Virzi (1992) proposed that three to five users would be enough to achieve acceptable results in industrial settings. This claim was based on studies where the cost of conducting one additional usability session (with one additional user) did not reveal much new information with respect to additional problem identification. As will be discussed later in detail, these studies did not include any considerations on the possible evaluator effect.

2.2.6 The facilitator and the evaluator

The literature on usability tests has not paid much attention to those conducting the studies (the facilitator and the evaluator) compared to the one who is being studied (the user). As in scientific experiments, one person is acting as head of the usability test, hereafter called the facilitator. The facilitator's role is to run the usability test, i.e., to prepare, introduce, conduct and wrap up each usability session. The evaluator is the person who is in charge of logging and analyzing data from the usability test. Typically, the evaluator communicates the results of the usability test to developers in a usability report or at meetings (Borgholm & Madsen, 1999). The facilitator and the evaluator could be one and the same person. However, the two roles are often distributed among two persons as some of the facilitator duties intervene with the duties of the evaluator (see also section 2.2.8).

2.2.7 The environment

A usability test can be run in a dedicated usability laboratory (Rubin, 1994) or in the field (Zirkler & Ballman, 1994), or in any setting in between these two ends. The laboratory is used to increase control of the usability test while attempting to retain the natural environment of an ordinary office or home, depending on where the product in question is to be used (see Figure 6, page 22). A traditional usability laboratory consists of a test room with office furniture, a computer, books on shelves, pin boards, posters, plants, and other items belonging to a typical office. More importantly, the test room is equipped with advanced video cameras and microphones that record user behavior. The evaluators and sometimes facilitators are hidden in a connecting room behind a one-way mirror from where they can observe the user. In the hidden room, evaluators and facilitators have access to video control systems enabling them to timestamp, tune, zoom, and in other ways refine the recordings.

Less advanced usability tests can be conducted on users' ordinary work place or in other less advanced environments. Moving the usability test into the field might increase the ecological validity of the study, but often it also decreases the control of the study. In a low-tech usability test, video cameras are sometimes skipped as recording medium. Instead, evaluators record usability problems on-the-fly. This recording methodology severely decreases the evaluator's ability to record user behavior precisely as the user does not stop up while the evaluator records problems. Hence, if two or more problems follow each other closely, only one of them might be detected and recorded. On the contrary, evaluators with great experience and stenographic techniques save much time recording on-the-fly compared to analyzing the session from a videotape. An alternative recording method is capturing activities on the screen electronically; this is similar to videotaping the screen, but rather than recording the activities on a videotape it is recorded on the hard disk of a computer. This method, however, does not record user verbalizations, and even with concurrent audio taping this recording method lacks contextual data and data about the user's emotions.

2.2.8 The procedure

The procedure for conducting usability tests consists of a general preparation phase, and a phase of running a number of usability test sessions. Each session consists of an introduction to the user, the actual test, and a debriefing of the users. In the preparation phase the usability team makes themselves familiar with the work environment in which the system is expected to be used. They interview, talk to, and observe users in their daily work settings, and perhaps set up workshops in order to get a deeper understanding for the system under evaluation. With the understanding of the work settings, through discussions with developers and after reviews of the system, the evaluator, facilitator, or other stakeholders set up appropriate task scenarios. Users are selected and invited to par-

ticipate after which the actual sessions can begin. If the facilitator is a member of the developing team, the preparation phase described here might already have taken place in the analysis phase of the development cycle. Thus, the facilitator might step directly into the actual usability sessions.

It is well known that users generally feel uncomfortable being observed in a test situation. They might be stressed by being in a strange environment, solving seemingly unproductive tasks while being observed by several people that they cannot see or hear. To diminish users' fear and stress, a good introduction is important for the success of the usability test session. Practitioners advise that users should be explicitly informed that they are not the ones undergoing a test – the system is. The user is informed that the more problems that are encountered during a usability test session, the more information is fed back to the development team. Another strategy advised by practitioners is to allow users to see all rooms in the laboratory including the hidden room and in other ways make the users comfortable before the session start.

In the introduction to the usability test, the facilitator should teach the user to think out loud. This is an unnatural thing to do for users and experience indicates that only few users are actually capable of giving valuable verbal reports about their work. With an extended introduction in thinking out loud this number should increase. The task scenarios should also be introduced to the user. If there is a time limit to each session, the number of task scenarios should not be exposed to the user, as they sometimes feel uncomfortable if they experience that they only get through a part of the task scenarios. The task scenarios should be presented with easy scenarios first and then followed by increasingly more advanced scenarios. Also, the introduction should inform the user about the communication form between the facilitator, the evaluator and the user. If the usability team does not intend to intervene in the session this should be specifically explained. Finally, the introduction should inform users that they are free to stop the session anytime they wish. For the matter of confidence and rights to use videotapes or other material, the user might be asked to sign a usability test session form with information on mutual agreements.

The actual test is initiated by reading the task scenario out loud, and handing over the written scenario to the user. If the task scenario is understood the session can begin. After finishing the first task scenario, the second scenario is presented in a similar manner, and so forth. When time runs out or the user has finished all task scenarios, the user should be debriefed. The debriefing aims to relax the user, talk about the task scenarios and session, and possibly get some insight into the system tested. Sometimes a session ends by interviewing the user or asking the user to fill out an after scenario questionnaire about the system (Lewis, 1991).

Having only one person acting as facilitator and evaluator at the same time might be problematic. For example, the facilitator/evaluator cannot adjust the video camera and simultaneously observe the user. However, if the analysis of the user is based on videotapes, it is less problematic then when observation and analyses is done on-the-fly.

2.2.9 The theory of verbal reports as data

The degree to which cognitive processes can be revealed by thinking out loud has received much attention in psychology and been investigated by Nisbett & Wilson (1977) and Ericsson & Simon (1980), amongst others. It is clear that verbal reports have limited scientifical validity in understanding basic cognitive functions. Ask a person "what is your mother's maiden name" and the person will, with little or no delay, answer the question (which can later be validated). Immediately after, ask the same person how she arrived to her answer and you will get no usable process data other than for example "I just remembered her name" (even with a more advanced explanation, this kind of data cannot be validated).

Usability tests do not aim for revealing basic cognitive processes. Neither do they aim for retrospective explanations for a given behavior. Thinking aloud in the context of a usability test is a convenient method to follow a user's plans, actions, and opinions. Verbalized plans are to help the evaluator to understand what the user is about to do, and why the user is clicking buttons or in other ways interacting with the system. Thinking aloud about actions being executed enables the evaluator to validate what the user is actually doing. Thinking aloud about user's preferences and opinions is, according to ISO/DIS 9241-11, an important aspect of usability and might lead to problem detection if users are frustrated about certain parts of the interface. Pairs of users working together while having a dialogue about their work has been suggested as more effective and more natural than thinking out loud (Hackman & Biers, 1992).

2.2.10 Logging and analyzing the data

The data from a usability test might consist of videotapes of the usability sessions, time measurements, and evaluator's notes recorded on-the-fly. Many sequential data analysis (SDA) tools exist on the market (Sanderson et al., 1994) with various possibilities for annotating clips of video or audio tapes or recording on-the-fly data in the analysis process. If the usability test has been conducted with logging on-the-fly as the only recording method, these notes exist as basis for the analysis. In practice, logging on-the-fly is an activity where data gathering and data analysis melt into each other. That is, an observation of a critical incident in a usability test situation is judged by the evaluator to be important enough to be recorded. This recording is only a representation of the situation – and probably only a weak representation, since only certain aspects of the situation are recorded. The representation of the situation is later reviewed in order to create a master list of problems. At this point the situation does no longer exist, only the representation of the situation exists. Thus, the data is based on an interpretation of a situation, which means that observation and analysis melt together when conducting on-the-fly usability tests.

Though this procedure is widely used in the industry to limit the costs of the evaluation and to rapidly communicate the results to developers, logging on-the-fly lowers the quality of data gathering and analysis for three reasons. First, when collecting data on-the-fly and often manually from an ongoing process, observation cannot be done while recording, and vice versa. Second, from the coarse-grained data material it is impossible to reconstruct events in order to get a better understanding of particular problems in the interaction between user and system. Third, from the data material alone it is not possible to see what are data and what is interpretation or analysis of data. Experience with data logging can possibly improve the overall data analysis, but it is not possible to completely eliminate any of the three mentioned problems when only logging data on-the-fly. Unfortunately, there does not seem to exist studies investigating quality differences between thorough analysis based on videotapes and on-the-fly analysis.

When has a usability problem been detected? Though there are different opinions on this question the most widely accepted answer is "when the user has problems accomplishing his or her task". More generally, if the evaluator judges something to be a usability problem it *is* indeed a usability problem. This definition either assumes that all evaluators detect identical sets of problems for the same usability session, or it implicitly admits that evaluators are different, thereby turning usability detection to a subjectively determined activity. Molich (1994) has described how evaluators can find usability disasters (rather than problems) when conducting usability tests. A serious interface problem exists if the user is unable to proceed without help from a human being, or the user experiences annoying, irrational behavior from the system. A user interface disaster is defined as if two out of at least three users experience the same serious problem. While the former definition (when the user is giving up) might be a quite reliable criterion, the later statement on user annoyance and

irrational system behavior seems unreliable as the problem detection depends on the evaluator's interpretation of the situation.

When an evaluator has recorded a problem description report for each identified problem from a number of usability test sessions, some problem reports might be doublets and some problem reports might contain more than one unique problem. Figure 7 shows an example of how individual user problem reports might be combined into a master list of unique problems.

The process of reliably constructing a master list of unique problems has received little attention in the literature (however, see Connell & Hammond, 1999; John & Mashyna, 1997; and Lavery et al., 1997 for explicit description of the analysis process). It is common practice in psychological experiments that judgments or classification of data is done by independent raters to delimit potential bias among researchers and to ensure that the data analysis does not invalidate the gathered data material. In a similar way, usability staff should aim at constructing a coding scheme for how to match and split up usability problem reports in order to enable more inspectors to produce lists of unique problems. These lists can then be compared for agreement among analysts in the matching procedure, and a consensus master list can be produced. The master list should also contain frequency information for each problem, i.e., a number for how many users that experienced a given problem.

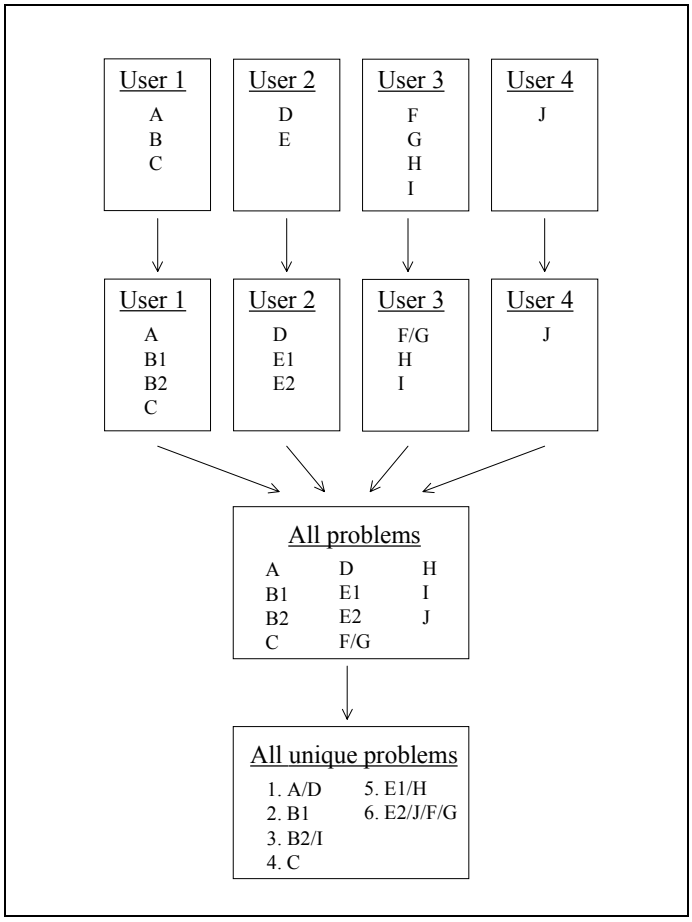

Figure 7. One possible procedure for creating a master list of unique problems based on four usability test sessions. The upper row represents the results of four usability test sessions. Each session produced a number of problems denoted as letters. Analyzing the problems leads to splitting up problem reports that essentially comprised more than one problem and combing problems that a user experienced more than once. The second row shows the result of this analysis. Problem B has been split into problem B1 and B2, problem E into E1 and E2, while problem F and G are essentially the same problem. In the third row all problems have been listed together in one initial master list. The last row represents the final master list of unique problems. Several of the initial problems were found to be doublets. Thus, the initial 10 problem reports comprise six unique problems.

The literature on usability tests has left minor attention to the analysis process. Papers describing usability tests focus on the preparation phase and the conduction of the actual test sessions rather than the analysis of the sessions. To ensure reliable results of usability tests, in particular in research studies, one has to collect data on videotapes, analyze these tapes based on rigorous usability criteria, and combine problem lists from different users to a master list of unique problems based on more raters.

2.2.11 Presenting the results

The result of a usability test is minimally a list of unique problems. This list should contain evidence on why a certain usability problem was detected (e.g., verbal reports or user action sequences), a free form problem description, and suggestions as how to correct the problem. Moreover, the problem list should contain information on how many users encountered a given problem

and a judgment of severity. It is a common practice to judge and report severity of problems in research reports. As we will describe later, judging problem severity is perhaps not as easy as one would think.

The usability problem report explaining the method used to identify the usability problems and the results of the test should be handed over to the development team (Molich et al., 1998). It has been suggested that the results should be communicated personally in a meeting with members from the usability team and developers to improve the persuasive power of usability evaluation. Also, developers seem to accept usability reports better if they get to see important clips from the videotapes in which important problems are revealed.

2.2.12 Summary

The aim of using usability tests is to identify usability problems in an interface in order to improve the usability of the system being tested. Any artifact with a user interface can be tested with usability tests. Simple systems can be evaluated fully, while complex systems are evaluated by selecting appropriate tasks scenarios. Users are selected randomly from a representative group of real users. Apart from the users, a facilitator and an evaluator participate in a usability test: the facilitator runs the study and the evaluator logs and analyzes the data. Usability tests can be conducted in controlled environments or run in real-life settings. Users are introduced to the usability test, they are presented with task scenarios, and they are asked to think out loud while going through task scenarios. The evaluators log or videotape the user and do further analysis based on the gathered data material. The result – a list of problems – is presented to the developers in order for them to correct the problems identified.

Usability tests conducted in real-life settings are generally limited by a number of factors. Limited resources often constrain usability teams to conduct usability tests with a small and not necessarily representative number of users. Moreover, short development cycles often force usability teams to communicate the results of the usability tests immediately after conducting the tests leaving only little time to analyze the data. Also, many companies are short of or completely lack employees with usability expertise. Finally, even when usability tests are conducted appropriately the results are not always used optimally (in fact they are sometimes completely abandoned) as other factors in the development process are viewed as more important than ensuring usability, e.g., fixing severe bugs, or finishing on time. All these constraints, of course, decrease the quality of usability tests conducted in real-life settings, although some argue that some usability work of limited quality is better than no usability work (Nielsen, 1993; Wright & Monk, 1991a & 1991b).

In research settings we need to apply rigorous methods to understand the strengths and weaknesses of UEM. Reliability and validity, for example, are important factors in research UEM studies, whereas it is viewed as less important in real-life settings. Therefore, we believe that usability tests – as an example of UEMs – could be used differently in real-life and in research projects. However, many parameters in usability tests have not been tested, as they are used in the industry, and hence, we do not know how reliable, how valid, how efficient, and how effective usability tests are the way they are used in real-life settings.

3 Learning and Using Cognitive Walkthrough

This chapter is an introduction to the case study paper in appendix 1, which has been submitted for publication to *Human-Computer Interaction* (Jacobsen & John, 1999). First we describe our motivation for investigating the learning process and usage of CW. We then describe two preliminary studies on learning and using CW. Finally, we briefly describe the background for the case study in appendix 1, which the reader is expected to read after reading this chapter.

3.1 Motivation for studying the learning process and usage of CW

At first blush, a that is UEM difficult to learn and use will be less attractive than UEMs that are easy to learn and use. However, as with achieving any skill, learning and using a difficult UEM might be a good investment, if the method turns out to be much more efficient than those UEMs that are easy to learn and use. In analogy, children spent hundreds of hours learning math or how to read and write. Most societies have accepted that these kinds of skills are worth the investment despite an extensive learning time. From the HCI world it is well known that fighter pilots put a lot of effort into learning how to fly airplanes. This training time is required to minimize errors and maximize performance. Even in ordinary offices, people spend much time learning how to use software running on computer systems. When usability staff apply UEMs to systems in the industry they are constrained by tight deadlines. Hence, usability staff require UEMs that are easy to learn and use. As long as no studies have concluded that a certain difficult-to-learn and difficult-to-use UEM is far better than other UEMs, practitioners will certainly prefer an easy-to-learn and easy-to-use UEM. The trade-off between ease of learning and usage versus effectiveness implies that we need to better understand how UEMs are learned and used. Another aspect is that we need to know how effective UEMs are, how they fit into the development cycle, and to what extent they can persuade stakeholders in the development process of the value of UEMs.

We have specifically focused on investigating the ease of learning and the first time usage of CW. We have made some preliminary studies on learning and using CW by means of qualitative methods (Jacobsen, 1996; Jacobsen, 1997). Moreover, in a case study we have examined two evaluators learning and using CW in order to get beyond the quantitative data. We have asked questions like *how* evaluators learn and use CW, *where* the evaluators have difficulties using the method, and *what* should be done to improve CW in the future (Jacobsen & John, 1999).

3.2 Preliminary studies on learning and using CW

Several studies on CW have preceded the study I have chosen to submit as a part of this thesis. First, as a novice with respect to CW, I conducted an introspective study (Jacobsen, 1996) in order to get a touch of the difficulties in learning and using CW. Based on my own experiences, I conducted another preliminary study testing whether it was feasible to skip over the tedious recording of success and failure studies in favor of only walking through the four questions for each action mentally. This work was conducted with a small sample of graduate students and was reported in Jacobsen (1997). Later again I investigated how system developers and users in real-life settings would learn to use and later apply CW to the project they were developing (Jacobsen, work in progress). In this section I will summarize the results of the two former studies which have been reported in unpublished manuscripts (Jacobsen, 1996; Jacobsen, 1997).

The starting point for my research in CW was an introspective study in which I read about the technique (in chronological order Polson & Lewis, 1990; Lewis et al., 1990; Jeffries et al., 1991; Polson et al., 1992; Wharton et al., 1994; Desurvire, 1994; John & Packer, 1995; John & Mashyna, 1997)

and conducted a CW of an e-mail system while recording my activities introspectively. The aim of this study was to get hands-on experience in using CW in order to outline a more detailed agenda for my research into UEMs. The findings of this study were reported in Jacobsen (1996) and will be summarized briefly below.

The interface under evaluation was Pegasus Mail for Windows. The fictive user description was specified with respect to the users knowledge about: 1) the domain (writing mail), 2) the operating system (Windows 3.1), 3) similar mail applications (e.g., Netscape Navigator) and the application under evaluation (Pegasus Mail for Windows). Three tasks were selected.

1) Read a new message and move the message under a different folder.
2) Copy 20 lines of a mail and send them as a new mail to a specific address.
3) Create an address book and a distribution list with two addresses and send a message to the distribution list.

The three tasks were transferred to three action sequences with a total of 69 actions. The action sequences were then walked through using the third version of CW described in Wharton et al. (1994). A total of 12 problems were identified in the CW; half of these problems were judged as severe problems for novice users. One severe problem was that the function for highlighting text did not follow the MS Windows standard. Another severe problem was that the function to create a distribution list included many features with no help in using those features; hence, the user would not be able to construct a distribution list. Five of the problems regarded mismatches between user's goals and label texts on menus and buttons.

The evaluation lasted 32 hours excluding the initial time spent reading about CW. The four activities taking up most time were rereading parts of Wharton et al. (1994) (not a part of the initial reading process) in order to apply the method appropriately, selecting task scenarios, setting up action sequences, and walking through the action sequences. In the first task, I spent 20 minutes per action; one third were spent on setting up the action sequence, and two thirds on walking through the action. The time spent per action went down to just below 10 minutes in the third task, and I did not expect this time to drop further with more experience in using CW. In order to get closer to the motivation for conducting CW in the future, I described my subjective opinion on my experiences and noted three problems with CW. First, as a total of nine hours passed by in the process of answering and recording the same four questions for the 69 actions, I found CW to be rather tedious to apply to an interface. Second, I found the recording of success and failure stories unnecessarily time-consuming; it especially felt unnecessary to record success stories. Third, CW felt less accessible to me as a novice evaluator than was promised in the literature, partly because of difficulties understanding at what level user goals should be described, and because the wording of the four questions is less than optimal.

Based on my introspective study, I conducted a small study in which I aimed to test whether CW could be taught to novice evaluators within a limited time, and if less recording in a CW process would relieve the tediousness of CW without deteriorating the quality of the method. This study was reported in Jacobsen (1997). The study was conducted with 13 graduate computer science students over a period of two weeks. In a double lecture (two times 45 minutes) the evaluators were introduced to CW and got hands-on experience with using the method. After the lecture, they were asked to read Wharton et al. (1994) and then participate in the study. Six groups of CW teams – with two or three evaluators each team – were formed. Each participant received a disk with the running system (a free-ware product called ClipMate for Windows), and two tasks to be analyzed. ClipMate is an advanced clipboard capable of holding several types of information simultaneously, enabling users to cut and copy text and figures in cascades for later cascades of pasting. The teams

were requested to evaluate two tasks. The first task (task A) was to open a text file and copy two particular words into the clipboard, then to open the program Paint and copy an element from Paint into the clipboard, and finally to copy these three elements from the clipboard into a word-processor document. The second task (task B) was to find four names in a text file, copy these into the clipboard and save these names in the clipboard, so that the names could be pasted into a program after the computer had been rebooted.

When using the original CW, the participants were required to document (1) action sequences, (2) the walkthrough (success and failure stories), and (3) problem description report (PDRs). In order to test the hypothesis whether CW would be just as effective with less recording requirement, I constructed an alternative CW, called ACW. Using ACW the evaluators only had to record (1) the action sequence for each task and (2) a PDR when a user interface problem was identified. Half of the groups were asked to set up an action sequence for task A and walk through this action sequence using the ordinary CW followed by setting up an action sequence for task B and walk through this sequence using ACW. The other half of the groups was asked to set up an action sequence for task A and walk through this sequence with ACW, followed by evaluating task B with CW. Besides handing in different types of recordings, the participants were asked to hand in documentation about their time consumption and to fill out a questionnaire about their background and their completed evaluation.

One important finding in this study was that all evaluators seemed to have acquired the skill of using CW in a limited time. I graded all solutions on a five-point scale: dissatisfactory, poor, satisfactory, good, and excellent. I graded all solutions to be somewhere between satisfactory and excellent with a large number of good solutions; this judgment was made by grading the quality of setting up action sequences, the recordings of success and failure stories, and the appropriateness of the evaluators' problem description reports. Moreover, the evaluators graded their own feeling of confidence with using CW after they had participated in the study; these self-estimates were in line with my grading of their performance. Another result of the study was that no differences were found in the use of CW versus ACW regarding the number of problems identified (collectively the evaluators detected 22 and 21 problem using CW and ACW respectively). Neither were there any significant differences in the evaluators' subjective opinion about the use of CW versus ACW. In fact, the evaluators liked both CW and ACW and found neither of the methods particularly tedious to use. The evaluators found that CW were perhaps more time-consuming than ACW, but also that CW seemed to increase the quality of the outcome compared to ACW.

A side effect of the study was that I found a significant difference between groups with respect to their problem lists. None of the groups constructed identical action sequences for the same tasks (ClipMate allowed the tasks to be performed in several ways). The number of problems identified varied from 4 to 12 problems between groups. And there was only a little overlap between the problems detected by different groups. One problem was detected by five groups, two problems were detected by three groups, and three problems were detected by two groups. The remaining 26 problems were not reported by more than one group. These results were surprising and suggest that CW is not as reliable as one might assume.

3.3 A tale of two critics: case study in using CW

The two studies described above were preliminary in the sense that I did not have much knowledge about CW when conducting the introspective study, while the sample size and setup for the study on CW versus ACW where inappropriate for publication. However, they both gave me important insights into CW, which has been essential for conducting and analyzing the case study of two

analysts learning and using CW (Jacobsen & John, 1999). The case study described in appendix 1 has been the most worth-while experience in my Ph.D. period both with regards to our findings but certainly also with respect to methodological insights into case studies (Yin, 1994). The motive for conducting a case study in CW is that there has been too much focus on quantitative work revealing insignificant conclusions with respect to understanding how CW is learned and used and why it produces the results it does. We have aimed at revealing those *how* and *why* questions through case studies.

Our case study paper comprises a background description, a description of the two cases, and a result section. However, the major part of the paper is devoted to describing qualitative differences between the two cases structured according to the four stages in learning and using CW:

1. Reading about CW
2. Defining user and choosing task scenarios
3. Transforming task scenarios to action sequences
4. Walking through action sequences.

Moreover, we have compared the two analysts' results with outcomes from a usability test, in order to reveal the predictive power of CW. In the conclusion we recommend changes in the CW procedures and suggest development of a computer-based tool to support the evaluation process. Now I expect the reader to skip to appendix 1, and read *A Tale of Two Critics: Case Study in Using Cognitive Walkthrough*, before continuing reading chapters 4 and 5 about the evaluator effect in CW and usability test and the discussion in chapter 6.

4 The Evaluator Effect in Cognitive Walkthrough

This chapter presents the study of the evaluator effect in cognitive walkthrough (CW). Due to the brief manuscript (Hertzum & Jacobsen, 1998) and limited space in the published paper (Hertzum & Jacobsen, 1999) more room is devoted in this chapter to describe the details of the study. Section 4.1 reveals results from previous studies on the evaluator effect. Section 4.2 focuses on the aim of pursuing research on the evaluator effect and in section 4.3 we present the method employed in our study. In section 4.4 we summarize the results and in section 4.5 and 4.6 we present the discussion and conclusion of the study. In the last sub-section we outline implications of the results and suggest future work on the evaluator effect in CW.

The results from the evaluator effect study on CW were presented in Munich in August 1999 at the eighth International Human-Computer Interaction Conference.

4.1 Evidence for the evaluator effect in CW

One empirical study on CW explicitly reveals the degree of agreement among evaluators (Lewis et al., 1990). In addition Hilary Johnson has given us access to the data set from Dutt, Johnson & Johnson (1994) that allowed us to reveal the evaluator effect in their study. The remaining papers on empirical studies of CW have not made it possible to reveal evidence on a possible evaluator effect (including Cuomo & Bowen, 1994; Desurvire, 1994; Ereback & Höög, 1994; Jeffries et al., 1991; Mack & Montaniz, 1994; Wharton et al., 1992).

Lewis et al. (1990) had four evaluators individually perform a CW of a simple voice mail system. The consistency across the four evaluators was fair in that more than half of the problems were detected by three evaluators but only one problem were detected by all four evaluators, see Table 3. The applicability of these results is however difficult to assess since three of the evaluators were familiar with the CE+ theory and had discussed the general trends of the data based on an empirical evaluation prior to their walkthroughs (see Lewis et al., 1990, pp. 238-239).

Dutt et al. (1994) had a student without HCI experience, a student with HCI experience, and a person with HCI research experience individually perform a CW of a personnel recruitment system. More than half of the reported problems were detected by all three evaluators, see Table 4, and a single evaluator found on average 73% of the total set of problems from the three walkthroughs.

Detected by exactly	No. of problems	
1 evaluator	2	(10%)
2 evaluators	5	(25%)
3 evaluators	12	(60%)
4 evaluators	1	(5%)
Total	20	(100%)

Table 3. The detected problems broken down by the number of evaluators detecting them, study by (Lewis et al., 1990).

Detected by exactly	No. of problems	
1 evaluator	11	(34%)
2 evaluators	4	(13%)
3 evaluators	17	(53%)
Total	32	(100%)

Table 4. The detected problems broken down by the number of evaluators detecting them, study by (Dutt et al., 1994)

Both Lewis et al. and Dutt et al. used the first version of CW described by Lewis et al. (1990) in their studies. Dutt et al. had made a number of changes to CW method suggested by Wharton et al. (1992) by 1) listing tasks in order of complexity, 2) advising evaluators to choose a higher level of actions than simple keystrokes when necessary, 3) advising evaluators to walkthrough repeated actions only once, and 4) advising evaluators to report problems they found even though they were not a part of the walkthrough process.

4.2 Aim of studying the evaluator effect in CW

Practitioners using UEMs expect that the methods are reasonable reliable. That is, two successive usability evaluation sessions should yield the same results. This definition of reliability only has theoretical value as testing such a hypothesis in practice fails due to evaluator bias. One evaluator using a given UEM to evaluate a given system will be biased toward the use of the UEM, the knowledge of the system and the results of the evaluation if she was to replicate the evaluation. Rather, a practical and interesting aspect of reliability is whether two different evaluators will produce the same problem lists when conducting an evaluation with the same UEM on the same system. We have called this reliability issue the *evaluator effect*. If a substantial evaluator effect is found in using CW, the method is less reliable than what we thought. Hence, practitioners will gain from this knowledge (perhaps by using other more reliable UEMs), while CW developers will have reasons to improve the method in order to make it more reliable.

The two studies mentioned in section 4.1 did not reveal a significant evaluator effect. However, three reasons still motivated us to carry through an evaluator effect study on CW. First, both studies were carried out using the first version of CW, and the current third version, described in Wharton et al. (1994), is quite different from the first version. Second, evaluator bias caused by prior discussions on usability test results partly invalidates the results revealed by Lewis et al. (1990). Third, the few evaluators involved in the two studies threaten the external validity of the studies.

Other motives for specifically investigating the evaluator effect in CW, are the results revealed in previous studies. In section 3.2 we described how we in one preliminary study – the one comparing CW to the slightly changed version of CW, called ACW – revealed results indicating the existence of an evaluator effect. Our case study in appendix 1 similarly revealed substantial individual differences among two evaluators. None of these studies, however, included a sufficient number of evaluators to firmly conclude that an evaluator effect actually exists in CW. Therefore, we wished to investigate this reliability issue in CW.

CW comprises several stages: getting to know the system to be evaluated, defining users, selecting appropriate task scenarios, transforming task scenarios to action sequences, and walking through action sequences in order to identify and report usability problems. In this study we wished to focus on the evaluator effect for those stages that are unique to the CW process, i.e., the transformation stage and the walkthrough process. Therefore, we purposely predefined task scenarios. If the study revealed a substantial evaluator effect for the transformation and walkthrough stages, we hypothesized that the evaluator effect for the overall use of CW (including getting to know the system, defining users, and selecting task scenarios) would be even greater.

4.3 The method employed in the CW study

In this section we present the method employed in the evaluator effect study on CW. We describe the system and the task scenarios, the evaluators, the procedure for the study, and the procedure for the compilation of the analyses of the evaluators' responses.

4.3.1 The system

In this study a prototype of a web-based library was used as the system evaluated (see Figure 8). The system, called HCILIB, gave the user access to a collection of scientific articles on human-computer interaction. HCILIB (Perstrup et al., 1997) integrates Boolean search with a scatter-gather inspired technique to display a browsable structure of the collection. Boolean searches can be expressed as conventional Boolean queries (using ANDs and ORs) or by means of a Venn diagram metaphor. The Venn diagram metaphor relieves the user from direct interaction with logical expressions. Instead, query terms are entered into two search boxes, A and B, and the search results are automatically sorted into three disjunctive sets corresponding to A–B, A∩B, and B–A (i.e., the user will always see all three sets as a result of a Venn diagram search). Each set result is annotated with a Venn pictogram indicating which of the three sets the collection belongs to. HCILIB is a quite small system intended for users with basic knowledge of the field of human-computer interaction and the World Wide Web. The evaluated version of HCILIB gave access to a collection of 135 articles from the ACM CHI conferences in 1995 and 1996.

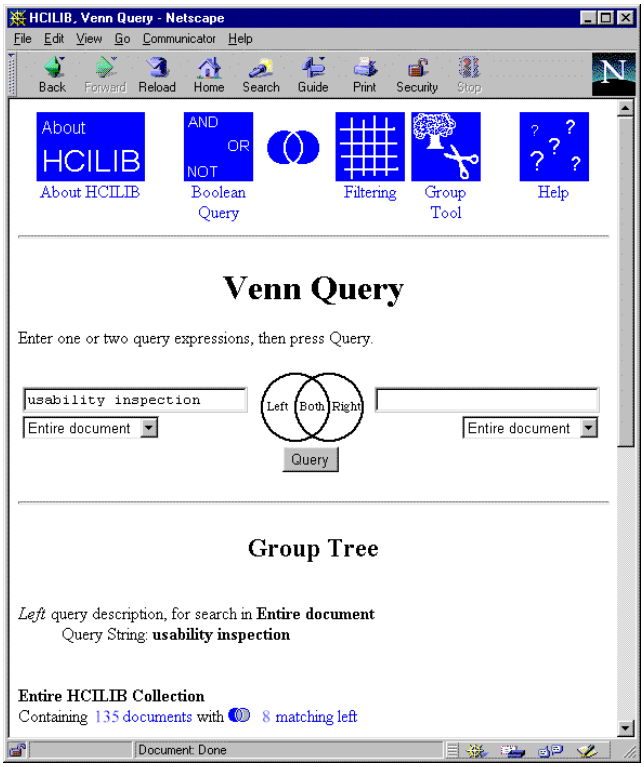

Figure 8. HCILIB's Venn Query page. The user enters a query phrase in the text box(es) and, possibly, limits the search to certain parts of an article by selecting for example title or abstract in the combo box, rather than 'Entire document'. The search is performed when the user clicks the Query button under the Venn diagram. Parts of the results of the search are displayed in the lower part of the page.

Three tasks were used in the evaluation. The two first tasks concerned simple and typical uses of HCILIB; the third task required a more thorough understanding of how the search facilities can be used to express more complex queries. The tasks appear below in Italics along with sample action sequences for solving them. The tasks were phrased exactly as written below, though in Danish. As the evaluators were to construct the action sequences themselves, the sample action sequences below are just examples of how they could possibly look like; all task scenarios could be transformed to different but still correct action sequences.

Task 1. Use HCILIB's Venn Query facility to find the articles concerning how the usability of computer systems can be evaluated without involving users. How many such articles are there? (In general, evaluation of the usability of computer systems is known as 'usability evaluation'. Evaluation that does not involve users is often referred to as 'usability inspection', while evaluation with users is called 'usability testing'.)

An example of an action sequence for task 1:

1. Click on the 'Enter the library' link to open HCILIB
2. Click on the 'Venn Query' button in the upper palette (as shown in Figure 8 the palette consists of six entries: "About HCILIB", "Boolean Query", "Venn Query", "Filtering", and "Group Tool". When one of the entries are selected the text in the palette disappears and the icon becomes inverted)
3. Move cursor to the left input field (this box does not automatically have cursor focus when the page is loaded)
4. Type 'usability inspection' in the box
5. Click on the 'Query' button below the inactive Venn diagram
6. Click on the link to the hit list, labeled '8 matching left'

Task 2. One of the articles that is found in task 1 is entitled 'Making A Difference - The Impact of Inspections'[5]. Bring this article up for closer study and make an electronic copy of it.

An example of an action sequence for task 2:

1. Click on the 'Document location' link of the article
2. Open the 'File' menu of the browser
3. Select 'Save as'
4. Enter a file name in save dialog box
5. Select a destination directory in dialog box
6. Complete the save dialog by pressing the 'Save' button in the dialog box

Task 3. The article in task 2 refers to an article on heuristic evaluation (a certain usability inspection method) authored by J. Nielsen & R. Molich[6]. Find all the articles that refer to this article on heuristic evaluation. How many such articles are there?

An example of an action sequence for task 3:

1. Click on the 'Venn Query' button in the upper palette
2. Move cursor to the left input field
3. Type 'Nielsen Molich heuristic evaluation' in the box
4. Click on the combo box below the input field
5. Select 'References' to restrict searching to the references at the end of the articles
6. Click on the 'Query' button below the inactive Venn diagram
7. Click on the link to the hit list, labeled '2 matching left'

4.3.2 The evaluators

The study was embedded in a grade-giving course where computer science graduate students were asked to construct action sequences for the three tasks above and do a CW of the tasks. The evaluators were taking an introductory course in human-computer interaction. Apart from their undergraduate degree in computer science the academic backgrounds of the evaluators included undergraduate degrees in business administration, chemistry, education, English, linguistics, physics, and

[5] Sawyer, P., Flanders, A. & Wixon, D. (1996) Making a difference - the impact of inspections, in Proceeding of the ACM CHI'96 Conference (Vancouver, Canada), ACM Press, 376-382.

[6] Nielsen, J. & Molich, R. (1990) Heuristic Evaluation of User Interfaces, in Proceeding of the ACM CHI'90 Conference, ACM Press, 249-256.

psychology. Half of the evaluators also worked part time as system developers in the industry. The evaluators were all male and their age ranged from 23 to 32 years with a mean of 25.5 years. The evaluators were experienced World Wide Web (WWW) users but they had no prior knowledge of the system being evaluated.

4.3.3 Procedure for the CW evaluators

Just before the assignment was handed out, Morten Hertzum, at that time Assistant Professor in computer science, gave the evaluators two hours of instruction in CW. This instruction consisted of a presentation of the practitioner's guide to CW (Wharton et al., 1994), a discussion of the guide, and an exercise where the evaluators got some hands-on experience and instant feedback. The evaluators had to perform their cognitive walkthroughs individually and to document them in a problem list describing each detected problem and the CW question that uncovered it. At a rough estimate, each evaluator spent 2-3 hours completing his CW. In addition, the evaluators produced a report describing their planning and execution of their walkthroughs, and their opinions on using the technique.

4.3.4 Procedure for the compilation of the analyses of the evaluators' responses

The eleven evaluators' raw problem reports comprised a total of 73 usability problems[7]. Based on a one page structured coding scheme two researchers (Morten Hertzum and I) created a master list of unique problems as follows. First, the two researchers split up those original problem reports that they thought contained more than one unique problem. I split 1 original report into two problem reports and 1 original report producing an additional 2 problem reports; Morten did not split up any original problem reports. Hence, both authors had 73 problem reports in common (with me having additional 3 problem reports).

Each author then examined their lists as to eliminate duplicates. Of the 73 problem reports common to the evaluators the researchers agreed on 60 (82%[8]) as to whether they were unique or duplicated. This calculation was compiled as follows: if both researchers agreed that one problem report was not matched to any other problem reports, this counted as an agreement in the problem matching procedure. If one of the researchers but not the other researcher matched one problem report with at least one other problem report, this counted as a disagreement. Finally, if both researchers matched a problem report to the same group of other problem reports, this was counted as an agreement.

For those problem matches that the researchers disagreed on, they made a specific proposition in order to reach consensus. There were no major disagreements in this consensus process of creating the master list as most initial disagreements were caused by differences in level of abstraction.

The final master list comprised 33 unique problems. The following examples of problems were identified: the query button in the center of Figure 8 could not be activated by hitting the enter button; users were not expected to find the correct query button, rather they would click on the pictogram above the query button; and, users would not notice that the results of a query had appeared on the screen after hitting the query button.

---

[7] In the papers in appendices 2 and 3 we mistakenly reported that there were 74 raw problem reports. Recounting the number of problem reports revealed that there were only 73. Note, however, that this mistake does not have any impact on the remaining analysis, as our analysis is solely based on unique problems rather than raw problem reports.

[8] Due to a different way of calculating the inter-rate reliability in the papers in appendices 2 and 3 the percents are different here. In the current summary we have made an effort to compile the results of the evaluator effect studies on CW and usability test similarly.

4.4 Summary of findings in the evaluator effect study

The evaluators differed greatly in how many problems they detected (see Table 5). The two evaluators that found the least problems had a detection rate below 10%, while the two that found most problems had a detection rate equal to or higher than 33% of the known problems in the interface. The actual data for the study is revealed in Figure 9. Each row in the figure represents an evaluator (in the same order as in Table 5) and each column represents a problem.

As much as 58% of the problems were detected by only a single evaluator, and no single problem was detected by all evaluators. A single evaluator found on average 5.9 problems (18%) of the known problems with a standard deviation of 3.20.

Evaluator	Number of problems detected	Percentage of problems detected
CW-E1	2	6%
CW-E2	3	9%
CW-E3	4	12%
CW-E4	4	12%
CW-E5	4	12%
CW-E6	5	15%
CW-E7	6	18%
CW-E8	6	18%
CW-E9	7	21%
CW-E10	11	33%
CW-E11	13	39%
Mean	5.9 (standard deviation=3.20)	18%

Table 5. Number of problems detected by all CW evaluators and the percentages of problem detection relative to the number of known problems.

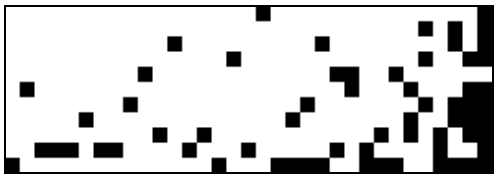

Figure 9. Matrix showing who found which problems. Each row represents an evaluator, each column a problem, and each black square that the evaluator detected the problem.

In this study, the most glaring finding is the diversity in the evaluators' problem detection. As much as 30 of the 33 problems were detected by at most three evaluators (see Table 6), suggesting a great deal of misses – or false alarms – in the performance of single evaluators.

Detected by exactly	No. of problems	
1 evaluator	19	(58%)
2 evaluators	8	(24%)
3 evaluators	3	(9%)
4 evaluators	0	(0%)
5 evaluators	1	(3%)
6 evaluators	1	(3%)
7 evaluators	0	(0%)
8 evaluators	0	(0%)
9 evaluators	0	(0%)
10 evaluators	1	(3%)
11 evaluators	0	(0%)
Total	33	(100%)

Table 6. The detected problems broken down by the number of evaluators detecting them.

Since single novice evaluators tend to only detect a minority of problems, we wanted to investigate how groups of evaluators performed compared to single evaluators. Figure 10 shows the average number of problems that would be found by aggregating the sets of problems found by different groups of evaluators (i.e., calculated as an average of all combinations for each group size). For each group a given problem was considered found if it was found by at least one of the evaluators in the group. We also fitted a curve to the data points based on the probability formula: $f(k) = n(1 - (1 - p)^k)$, where n is the estimated number of problems found in the interface given the three tasks, p is the probability for an evaluator to identify a problem and k is the number of evaluators. For the fitted curve the following numbers applied, n = 43, p = 0.121, and k = 11. The squared correlation coefficient were high (R^2=0.998) with a standard error for n estimate = 1.4%. However, the standard error for p estimate that has been reported in appendices 2 and 3 does not apply in this setup, as it later became apparent to us that the data points are not independent from each other (averaging over all possible combinations re-uses data points and hence violates the assumption that data points should be independent of each other).

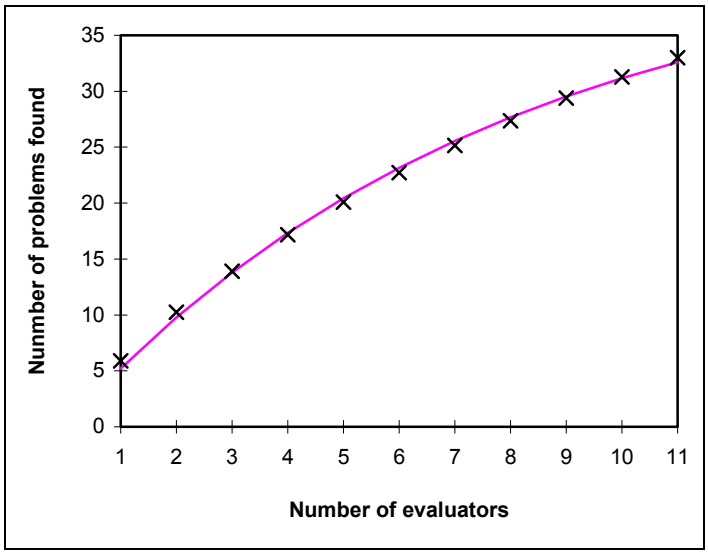

Figure 10. The number of problems detected for all possible group sizes comprising individual working evaluators. The data points are the aggregated values for all possible permutations for that group size. The curve plots an estimate given by the theory of probability formula $f(k) = n(1 – (1 – p)^k)$, where n = 43 (estimated total number of problems), p = 0.121 (probability of identifying a problem for one evaluator), and k = 11 (the number of evaluators).

4.5 Discussion

With the results described above two questions are raised. First, why did our results differ from those revealed by Lewis et at. (1990) and Dutt et al. (1994)? And second, why did the evaluators produce so different problem lists in our study? In the next two sub-sections we will try to answer these two questions.

4.5.1 Comparing our results to previous studies on the evaluator effect

Why did the results from our study reveal a far larger evaluator effect than the results from the study by Lewis et al. (1990) and Dutt et al. (1994)? We have investigated two plausible causes for this difference. The first hypothesis regards our procedure in compiling the results. If the process of matching the evaluators' problem reports to a master list of unique problem tokens caused us to produce more unique problem tokens, it might be a reasonable explanation for the differences. The other hypothesis is of a methodological nature. If there is a negative correlation between number of evaluators and the size of the evaluator effect this could also explain the differences between the results from our study and those of Lewis et al. and Dutt et al. In this section we will try to confirm or reject these hypotheses.

The first hypothesis regards whether changing the procedure for compilation of raw problem reports and creation of a master list of unique problem tokens will change the evaluator effect results or not. In our study two investigators independently created two master lists. Then they discussed those problem matches that they did not initially agree on in order to archive consensus. An alternative approach of handling the disagreements is to consistently apply the following rule. If one investi-

gator judges problem *a* and *b* to be the same and the other investigator judges *b* and *c* to be the same, then problem *a*, *b*, and *c* should be grouped together (i.e., they should represent one unique problem token on the master list). The result of applying this rule to the disagreements between our two investigators in their matching procedure is fewer problem tokens on the master list. A single evaluator detected on average 21% of the 26 problem tokens defined by this procedure. Thus, not much different than the 18% reported in section 4.4. In the study by Dutt et al. (1994) the grouping of the raw problem reports were split into two stages: matching the problems reported by individual evaluators into an intermediate master list and generalization of common problems in the process of creating the final master list. One investigator compiled their results. A single evaluator detected on average 65% of the 45 problems on the intermediate master list and the aforementioned 73% of the 32 problems on the final master list. Thus, the results do not seem to be particularly sensitive to changes in the grouping of the problem instances. Based on the paper of Lewis et al., we do not know exactly how they constructed their master list.

The other hypothesis is that the number of evaluators and the degree of an evaluator effect are negatively correlated. That is, the fewer evaluators in an evaluator effect study the higher detection rate per evaluator and thus the higher correlation between different evaluators. The interesting part of this hypothesis is the method for calculating the detection rate rather than the relation between number of evaluators and the degree of the evaluator effect. The detection rate for a given evaluator is typically calculated as follows: number of problems detected by the evaluator divided by known problems. However, the accuracy of the calculation of the detection rate depends, in turn, on whether the evaluators collectively detected all problems in the interface. To illustrate the relationship between the number of evaluators and the calculated detection rate our study can be scaled down to studies with fewer evaluators. This is done in three steps.

1) Selecting *x* number of the eleven evaluators.
2) Counting the number of unique problems *x* evaluators have collectively detected (i.e., detected by at least one of the *x* evaluators).
3) Calculating how many problems a single of the *x* evaluators detects on average of the problems detected collectively by the *x* evaluators.

Table 7 shows the average number of problems obtained by repeating this process for all possible combinations of evaluators. Thus, we find that three of our evaluators collectively detect 13.91 problems on average for all combinations of three evaluators and hence a single evaluator detects on average 43% of these 13.91 problems. With only four evaluators in our study the average detection rate goes down to 35%. In comparison one evaluator of the three evaluators participating in the study by Dutt et al. (1994) detected 73% of the problems on average and one evaluator of the four evaluators participating in the study by Lewis et al (1990) detected 65% of the problems on average.

No. of evaluators	Total no. of problems found	Percentage of problems found by one evaluator
1	5.91	100%
2	10.24	59%
3	13.91	43%
4	17.16	35%
5	20.08	30%
6	22.73	26%
7	25.15	24%
8	27.36	22%
9	29.40	20%
10	31.27	19%
11	33.00	18%

Table 7. Scaling our study down to fewer evaluators. For each number of evaluators the table shows the number of problems they collectively find on average for all permutations and the percentage of these problems that one evaluator on average detects.

The seemingly increase in the evaluators' performance as the number of evaluators in the study decreases is an artifact of the assumption that the number of problems found collectively by the evaluators is the same as the total number of problems in the interface. In our study a group of three evaluators is not enough to find the majority of problems. Moreover, a group of three evaluators is not enough to reliably predict the proportion of the problems detected by a single evaluator. In general, detection rates from different studies are not comparable unless they are based on the, if necessary, estimated total number of problems in the interface. And even when this premise has been fulfilled there will be differences between studies toward regards detection rate and evaluator effect due to different systems tested, different evaluators, different settings, etc.

In summary, changing our matching procedure from being driven by consensus discussions for those problems that we matched differently to a more rigorous but also more rigid method did not change our overall results much. However, calculating detection rates and the evaluator effect based on a smaller sample of evaluators tend to overestimate the average detection rate for a single evaluator. Hence, calculating the detection rate based on known problems in an interface is wrong. Rather this calculation should be based on *total* number of problems, a number typically estimated. A further practical issue is that such calculations are not reliable with too few evaluators. In fact, in our study we expect there will be more than the estimated 43 problems for the tasks tested, although this expectation is based on intuition rather than facts (see the later section 6.3 for an alternative view on the definition and detection of problems).

4.5.2 Plausible reasons for the evaluator effect

We have found two plausible reasons for the evaluator effect in CW: stereotyping and anchoring. The walkthrough process – answering the four questions for each action in an action sequence – is based on a fictive user description. If the evaluator is following the suggestions from Wharton et al. (1994) this description should be general and rather brief. The user description does not by itself bring any assumption about individual differences among users into the walkthrough process; in fact it is a description of a *single* user. Therefore, the evaluator will probably stereotype the simulated user behavior giving little possibility for including various behaviors from the possible heterogeneous user population. The stereotyping of users does not necessarily result in any evaluator effect, as evaluators might simulate the same stereotyping in their evaluation process. What might cause the evaluator effect is the lack of guidance in the fictive user description, which forces the evaluators to judge a not-well-described user's behavior for a particular action. The most spontaneous solution for the evaluator is to think of one's own behavior for the particular action in question, i.e., anchor the answer to the question to the evaluator's own experience. Here is where the evaluat-

ors might break away from each other. The evaluators are different, and hence their anchoring strategy might lead them to different answers even for the very same action.

We investigated the stereotyping and anchoring hypotheses by looking closer into how the evaluators answered the four questions on identical actions. Though the evaluators constructed their action sequences from the same three predefined tasks only 4 out of an average of 15 actions were identical across all evaluators. One of these actions is to execute a query by activating the Query button shown in Figure 8 on page 38. In walking through this action, three evaluators reported success stories on all four questions, while eight evaluators reported a total of five different problems. Three evaluators reported that the user would click a Venn pictogram situated above the Query button, rather than the button itself. Three evaluators reported weak feedback from the system after clicking the Query button. Two evaluators reported that the Enter key does not execute the query, i.e., the user has to use a pointing device to activate the button. One evaluator reported that the caption on the button should be changed. And finally, one evaluator reported that the user would forget to activate the Query button. It seems quite reasonable that all problems would actually happen for some users in a real situation, just as some users might experience no troubles using the Query button, as suggested by three evaluators. Though all evaluators' use of the four questions on the analyzed action seems reasonable, the outcome is very different across evaluators. The same pattern was found for the three other actions that were identical across the evaluators.

The differences among evaluators with regard to answering the same questions for identical actions might also be attributed to other reasons than anchoring and stereotyping. However, the case study on the two CW analysts (A1 and A2) described in Jacobsen & John (1999), appendix 1, supports the hypothesis on the anchoring behavior. Several of the problems revealed by A1 and A2 were identified as problematic by the analysts long before they conducted the actual CW analysis based on a fictive user. Future studies should be conducted to confirm or reject especially the stereotyping hypothesis.

4.6 Conclusion

This study investigated 11 novice evaluators conducting CW analyses on three set tasks on a web-based library system. They detected a total of 33 problems. A single evaluator detected 18% of the known problems on average, but estimating the total number of problems for the tasks analyzed (43 problems) the detection rate was as low as 12%, and the same estimate further suggested that as many as five evaluators would be needed to detect half of the estimated problems.

The evaluator effect is not unique to the CW technique. Other inspection methods like heuristic evaluation are known to have the same property (Nielsen & Molich, 1990), although we assumed that CW would be more reliable than heuristic evaluation given the rather structured process of conducting CW evaluations. We have hypothesized that CW has a special weak point as the description of the fictive user prompts evaluators to stereotype the user group and lacks guidance on how to simulate the users behavior thereby introducing an anchoring effect of the answers to the evaluators' own experience.

Our study was an initial investigation of the evaluator effect in CW based on novice evaluators. Several methodological limitations restrict the generalization of our results. First, the results are restricted to the system tested – a web based on-line library. Second, we did not test the individual differences in the construction of task scenarios as we purposely wished to understand the processes particular to CW, namely the transformations of tasks to action sequences and the walkthrough of

the action sequences. Third, the evaluators' learning process before using CW was restricted in time and was not controlled for success before the evaluators participated in the study[9]. Finally, we did not have any control over the evaluators work (i.e., under which circumstances they conducted their evaluation) and though the course that the evaluators followed were grade-giving, we could not control their degree of motivation. As a counterbalance for these restrictions we had more evaluators conducting the CW than what has been presented in previous papers on the use of CW.

4.7 Implications and future studies

Despite the restricted results in our study we believe they have implications for practitioners, researchers and those who intend to further improve CW. Also, because the study was restricted in various ways we will suggest specific future studies on the CW in this section.

CW does not seem to be reliable when novice evaluators transfer tasks to action sequences and walk through those action sequences in order to produce problem lists. Therefore, in real-life settings more evaluators should probably perform cognitive walkthroughs on the same system to increase the reliability of the results. To take it to its logical conclusion CW is perhaps not sufficiently mature to meet the demands of practitioners due to low performance for an individual evaluator and the lack of reliability. Future studies have to address whether longer training period and more experience among evaluators in the use of CW will decrease the evaluator effect.

Researchers comparing UEMs or those who perform studies on particular aspects of CW should also take the evaluator effect into account by including more evaluators in their studies. Finally, CW developers and others should conduct further studies on the anchoring and stereotyping hypotheses. Until our study has been replicated we encourage CW developers to consider how the fictive user description could possibly contain descriptions of more than one user to take into account the typical heterogeneous user group of a real system.

[9] The researcher, who taught the evaluators how to use CW, had followed a lecture in the use of the method three years before this study was conducted. Moreover, he had applied the technique to a patient data management system. In the preparation to this study he reread several papers by the CW developers and have had extended discussions about the method with the other researcher of this study.

5 The Evaluator Effect in Usability Test

This chapter is devoted to the evaluator effect in usability test. In section 5.1 we describe previous work on the evaluator effect and user effect in usability test. Section 5.2 presents our motivation for studying the evaluator effect. In section 5.3 we describe how we designed the study on the evaluator effect. Section 5.4 summarizes the results of the study, while section 5.6 concludes the study. Section 5.7 describes implications and suggests future work.

This study was first presented in Los Angeles at the ACM CHI'98 conference and published in the conference proceedings as a short paper (Jacobsen et al., 1998a). Later we elaborated on the study by collecting data on evaluators' ability to judge problem severity. The analysis and results of this study was presented in Chicago at the Human Factors and Ergonomics Society 42nd Annual Meeting as a full paper (Jacobsen et al., 1998b).

5.1 Evidence for the user and evaluator effect in usability test

The evaluator effect has been either neglected or unknown to researchers investigating aspects of the usability test. The user effect, on the other hand, has been investigated by more researchers in the field with results that to a great extent have been transferred to the industry. Researchers' motivation for studying the user effect was to decrease the number of users without deteriorating the quality of the studies in order to save critical time in the development cycle.

The next two sections reveal previous work on the user and evaluator effect in usability test. The user effect is described based on empirical studies, while the evaluator effect is described from a theoretical perspective due to the lack of empirical evidence.

5.1.1 The user effect in usability test

The number of subjects in psychological research experiments is often set to a level appropriate for analyzing data statistically. Usability staff do not typically aim for statistical analysis (although they sometimes do when conducting validation usability test), and hence they might decrease the number of users compared to research experiments. Considering individual differences among users we know that studying one user is not sufficient to capture the majority of problems in an interface. So, how many users are enough? Lewis (1994), Nielsen & Landauer (1993), Nielsen (1994), and Virzi (1990, 1992) have all investigated how many users are sufficient when running usability tests in industrial settings. Based on his first study Virzi anticipated the number of users required to be no more than 10, which he adds "*may be substantially fewer subjects [=users] than would be used in a 'typical' usability evaluation*" (Virzi, 1990. p. 293). Based on three different experiments described in his later paper he found that only five users were necessary to capture 80% of the known usability problems (Virzi, 1992, p. 467). Based on five usability tests Nielsen & Landauer found that they needed between four and nine users (with an average of five) to find 80% of known usability problems. Nielsen's final recommendation based on two more experiments in his later paper were "*to plan for 4 $\pm$ 1 subjects in a thinking aloud study*" (Nielsen, 1994, p. 393).

Generally, all researchers make use of the same probability formula to predict problem discoveries in usability test. The proportion of problems found with n users = $1 - (1 - p)^n$, where p is the probability of finding the average usability problem when running a single, average user. This formula is only an approximation to the real world, as more of the assumptions for using the formula cannot be taken for granted. One assumption is that the probability of finding any given usability problem in any given session is independent of the outcome of previous sessions. This assumption is not met,

since evaluators are biased by problems identified in previous usability test sessions. Another assumption is that all usability problems are equally easy to identify. This assumption is – for obvious reasons – not met either. Although these assumptions are not met in real-life settings (or in extremely controlled environments) the formula has shown to model the outcomes of several usability tests surprisingly well. According to Nielsen & Landauer (1993) and Lewis (1994), the formula can be used in practice to estimate the number of users necessary to identify for example 80% of the usability problems; at least after the evaluators have run a few sessions in order to calculate the probability of finding one usability problem for that particular interface.

In both research and practice it has been widely accepted that as few as 3 to 5 users were sufficient to capture the majority of problems. However, researchers disagree with respect to whether problem severity is associated with identification of problems. Virzi (1992) found that more severe problems are easier to detect than less severe problems. This finding is interesting from a qualitative point of view, as usability staff want to identify severe problems rather than all problems. One severe usability problem might be 100 times more important to fix than fixing 100 minor problems. Running fewer users in a usability test saves time and money. And if this procedural change has no negative impact on the detection of severe problem we have a double-win situation. Nielsen (1992) found the same positive correlation between problem detection and problem severity although not as dramatically as in Virzi's studies. Lewis (1994) however, failed to replicate a positive correlation between problem detection and problem severity in his independent usability test.

The whole issue on the user effect might be more complex than expected. How does a possible evaluator effect impact the results of the user effect revealed by researchers in the field? Moreover, there seems to be disagreements on whether problem identification and problem severity is correlated, and we might need more investigation on this matter. In section 5.4 we will reveal the outcomes of our studies regarding the evaluator effect and its impact on the results of user effect studies. Moreover, we will discuss our view on the correlation between problem detection and problem severity based on our results on this issue.

5.1.2 The evaluator effect in usability test

We have not been able to find any empirical evidence for the existence of an evaluator effect in usability test prior to our study. Those using or investigating usability tests typically only make use of one evaluator and a number of users. However, it was not too difficult to find non-empirical evidence for the evaluator effect presented in psychological literature as well as in HCI papers although to a letter extent. Dedicated teaching material – like text books in HCI – generally include a section on UEMs. These books, however, do not reveal any precautions of a possible evaluator effect.

The reliability and validity of UEMs has been investigated by Wenger & Spyridakis (1989). Based on behavioral and psychological literature they identify two types of reliability and validity problems when using usability tests, namely problems related to direct observation of users, and problems related to ratings of verbal protocols.

Direct observation is considered as one of the most precise data gathering techniques compared to verbal protocols, surveys, questionnaires, interviews, etc. (Cone, 1977). However, in order to rely on direct observations, observers need to predefine the subject's behavior of interest in details. Most often usability evaluators (as observers) do not specify exactly what they are looking for, other than "usability problems". Without defining a usability problem in concrete terms evaluators make their first mistake when relying on direct observation. Even when observers have defined the area of interest in concrete terms, inter-rate reliability among observers should be measured (Holleran, 1991).

Wenger & Spyridakis (1989, p. 268) are in line with this suggestion, "*The degree of agreement between observers should always be checked*". Setting up concrete usability criteria and calculating inter-rate reliability of the results are not applied in any of the usability test papers we have surveyed. Although direct observation is highly rated compared to other data gathering methods there are pitfalls using this method. Scarr (1985) notes that humans tend to look for facts that support their prior beliefs, and based on empirical evidence Brewer & Chinn (1994) argue that individuals' beliefs in a theory strongly influence their evaluation of the data. There is no doubt that developers (and even so-called objective usability specialists) acting as observers in a usability test might have ideas and theories about the success of the tested product that might interfere with their observation.

Verbal reports are often added to direct observations in usability tests. Verbal reports cause problems for evaluators because of the potential for ambiguous interpretation of the protocol data and the sheer volume of data generated. Typically a key or coding scheme is constructed for easy classification of the subjects' reports, but in classical, psychological experiments disagreement among observers' classification of verbal reports is a well-known phenomenon. Training observers in classifying data, which would be the spontaneous solution to this problem, has sometimes shown to even decrease the accuracy of the scores (Cone, 1977; Bernadin & Pence, 1980; Borman, 1979; Lord, 1985). Even though Ericsson & Simon's (1980) extensive paper on verbal reports gives suggestions on how to delimit the possible investigator effect of collecting verbal reports, this piece of work is not used actively by researchers and practitioners (although citations of Ericsson & Simon (1980) are often seen in usability test papers and books).

Although psychological literature comprises information on deficits in using direct observations and verbal reports, we have not been able to find warnings in HCI text books on a possible evaluator effect. In usability tests, as defined earlier, users are typically asked to think out loud and the sessions are videotaped for further analysis. In textbooks and papers describing usability test, not much is said about how to transform observation of users to data or how to encode these data (for missing discussions on these issues see e.g., Jordan (1998), Shneiderman (1998, pp. 127-132) Dix et al., (1998, pp. 427-431), and Preece et al. (1996, pp. 619-621). Nielsen (1993) has discussed reliability issues in usability test, but he only states that users are different without raising the question on differences among evaluators/observers. Virzi states that "*The think-aloud method incorporates the users' perspective directly, without being filtered by an intermediary, and can be cost-effective under some circumstances*." (Virzi et al., 1993). These particular papers and books are just typical examples of HCI literature where the evaluator effect has not been considered. This lack of information in HCI literature regarding a possible evaluator effect stands in sharp contrast to the methodological knowledge among observers obtained from psychological theory and practice.

5.2 The aim of studying the evaluator effect in usability test

When practitioners evaluate user interfaces using usability test rather than expert review, heuristic evaluation or another inspection method, it is primarily to ensure the quality and persuasive power of their evaluation work. They expect that usability tests are reasonably reliable. One motivation for investigating the evaluator effect in UEMs is to inform practitioners about a particular reliability issue, the evaluator effect. If usability tests show minor evaluator effects, we might hypothesize that it is reasonably reliable regarding the evaluator using the method. On the contrary, if we find big differences in the outcomes of evaluators analyzing the same usability test sessions it indicates that the outcomes of usability tests are more dependent on evaluators than what was initially expected.

Another aim for studying the evaluator effect was to put user effect studies into perspective. More researchers have presented empirical and theoretical evidence suggesting that three to five users

would be sufficient to detect the majority of problems when applying usability test to an interface. These studies did not consider a possible evaluator effect, and hence the number of problems in an interface might be dependent on both number of users and number of evaluators. In their aim of constructing a mathematical model of the finding of usability problems, Nielsen & Landauer (1993) assumed that the *k* in the probability model could be both *users* in usability tests and *evaluators* in heuristic evaluation. Hence, they assumed that the effect was similar for these different groups of participants using different UEMs. This assumption might be wrong.

Our motivation for investigating the evaluator effect was not only to inform usability practitioners and to put user effect studies into perspective, but also to challenge the methodology employed in comparative UEM studies. A considerable number of comparative studies of UEMs have been published recently (Bailey et al., 1992; Connell & Hammond, 1999; Cuomo & Bowen, 1994; Desurvire et al., 1992; Hammond et al., 1984; Henderson et al., 1995; Jeffries et al., 1991; Karat et al., 1992; Karat, 1994; Smilowitz et al., 1993). Generally these studies have compared number and types of problems identified using two or more UEMs. In some of these comparisons, usability tests have been used as a yardstick for other UEMs without knowing about or being concerned about the possible evaluator effects. If the evaluator effect is substantial for any UEM, then "between subjects" comparative studies of UEMs might say more about the evaluators using the UEMs than the techniques themselves. Hence, another motivation for studying the evaluator effect is to put comparative UEM studies into perspective.

Studies revealing actual evaluator effects will not only inform practitioners and those researchers interested in conducting user effect and comparative studies. They will also be a contribution to the overall knowledge of usability tests both in practical usage and in light of future research.

5.3 The method employed in the evaluator effect in usability test

This section describes the system used in our study, the user, the facilitator, evaluator profiles, and the procedure for conducting the study and compiling the analyses of the evaluators' responses. The method section presented here contains more information than in the method sections in the published papers (appendix 4 and appendix 5) due to limited space in the published papers.

5.3.1 The system

The system used in our study was a multimedia authoring system hereafter called the *Builder* (Pane & Miller, 1993). The Builder enables college and university teachers to build computer based learning material for science students. The Builder runs on Macintosh machines. A teacher can build so-called *volumes* that runs on computers. Typically a volume consists of plain text, high-resolution pictures, movies, animations, slide boxes, and small pieces of programming code. Most of the items in a volume are locked, in the sense that they are not editable by students; however small pieces of programming code in a volume can for example be introduced to let the students play with variables that are linked to an animation. The users in our study represented those who are intended to build a volume for students, rather than students using volumes. That is, users in the study should not use a volume to learn something about a science topic. Rather, they were asked to build a volume that could be used by students to learn about a science topic, in our case a topic in biology.

5.3.2 The users

Four experienced Mac users were asked to participate in the study for approximately one hour each. The users worked individually. None of the users had previous experience creating volumes in

Builder. The users were asked to create a new volume based on a printed target document. Five task scenarios were constructed in order of complexity. The task scenarios consisted of creating several pages in the volume, adding and editing some glossary items, adding entries to a table of contents, deleting a page, switching two pages, and saving the volume in two different versions. The users did not receive any instructions in the use of the Builder prior to the usability test. In the following, all users will be addressed as "he" though both males and females participated.

5.3.3 The facilitator

The same facilitator ran all four usability test sessions. She knew Builder in details and had previous experience with usability test sessions. The facilitator videotaped all sessions so that the videotape captured the full screen used in the study, but not the keyboard, the mouse, or the user. A microphone attached to the user ensured high quality recordings of the users' verbal reports. To enable comparisons between usability sessions, the facilitator used the same procedure (which was carefully written down) for each session. The facilitator did not interrupt the user unless the user explicitly gave up completing a task, forgot to think out loud, the system crashed, or three minutes passed without any progress in solving the task scenarios.

5.3.4 The evaluators

Four HCI research evaluators participated as evaluators. All evaluators were familiar with the theory and practice of the usability test. Two of the evaluators had extended experience analyzing usability test sessions, while the remaining two evaluators had less experience. The evaluators' initial experience with the system (the Builder) also varied (see Table 8). Two evaluators had experience with the Builder before this study was initiated (E1 and E2), while the two other evaluators were asked to familiarize themselves with the system prior to the evaluation. As prior experience with the Builder had been an uncontrolled factor for E1 and E2, we did not wish to control (although we measured) how much time E3 and E4 spent to feel comfortable with the system before the actual evaluation[10]. Three of the four evaluators were also authors of the published papers (E1, E2, and E3). As E1 and E3 had no input into the design of the usability test they later conducted the compilation and analyses of the evaluators' responses (see later how results where blinded in the analyses process). In the following, all evaluators will be addressed as "she" though both males and females participated.

Evaluator	Occupation	Number of users previously analyzed	Initial experience with the Builder	Average analysis time per tape
E1	Associate professor	52 users	10 hours	3.8 hours*
E2	Doctoral student	4 users	5 hours	2.7 hours
E3	Assistant professor	6 users	2 hours	2.9 hours
E4	Usability lab manager	66 users	12 hours	4.5 hours

Table 8. The HCI research evaluators' previous usability test experience, their experience with the Builder and the average time spent analyzing each tape (each tape lasted approximately 1 hour). *The analysis time shown for E1 is the time spent analyzing the last tape, as she did not keep track of the time she spent on the first three.

5.3.5 Procedure for the usability test

The facilitator introduced the users to the study. Originally, the usability test was conducted to find usability problems in the Builder interface, and the facilitator was unaware that a later evaluator effect study was established based on the videotapes from the usability test. The facilitator intro-

[10] E2 and E3 ended up having spent the most and the least time respectively familiarizing themselves with the system.

duced the users to the thinking aloud procedure and he ran a pilot test on a computer game where he asked the users to think aloud while playing the game. The facilitator then told the user about the roles played by the facilitator and the user. The facilitator sat in another room throughout the session but he watched the session through a slave monitor wired to the video camera. If the user forgot to think out loud, the facilitator would interrupt the user to remind him about the thinking aloud procedure. If the user had troubles solving a task, the facilitator would not help the user unless the user explicitly gave up on the specific task. Hence, the user was encouraged to keep trying for possible solutions rather than giving up. However, if more than three minutes passed without any improvement, the facilitator would interrupt the user to keep moving along in the task scenario. The first task scenario was read out loud by the facilitator and was handed over to the user in writing. If the user succeeded solving the first task, the next task scenario was handed over, unless the overall time limit of an hour per session had been overrun; and so forth with all tasks. The facilitator debriefed the user after each session asking about the user's opinion on the system and the session.

The evaluators based their evaluation on the four videotaped sessions, each containing one user thinking out loud while going through the set task scenarios. Throughout the analysis the evaluators also had access to a written specification of the Builder (35 pages) and the running system. The evaluators were asked to detect and describe all problems in the interface based on analysis of the four videotapes in a preset order. No time limits were enforced. To minimize an evaluator effect caused by different usability problem definitions and reporting style among evaluators, we predefined those conditions. Moreover, all evaluators were aware that their resulting problem lists would be compared to each other, which probably motivated them further to do their best. The evaluators were asked to use the following set of usability criteria:

(1) the user articulates a goal and cannot succeed in attaining it within three minutes
(2) the user explicitly gives up
(3) the user articulates a goal and has to try three or more actions to find a solution
(4) the user creates an item in his new document different from the corresponding item in the target document
(5) the user expresses surprise
(6) the user expresses some negative affect or says something is a problem
(7) the user makes a design suggestion
(8) the system crashes
(9) the evaluator generalizes a group of previously detected problems into a new problem.

The evaluators were requested to report five properties for each detected problem:

(a) the user videotape analyzed
(b) a videotape time stamp for each detected problem (automatically recorded via a SDA (Sequential Data Analysis) tool, called MacSHAPA)
(c) evidence consisting of the user's action sequence and/or verbal utterances
(d) one of nine predefined criteria for identifying a problem
(e) a free-form problem description

Table 9 shows two examples of problem reports following the procedure described above; the evaluator recorded user videotape number, time stamp, evidence for the problem, usability criteria used, and the evaluator's free form problem report.

Eva-luator	User	Time stamp	Evidence	Usability criteria	Free-form problem report
E1	1	0:40:41	User says, "*no, wrong size…no, doesn't fit either…why is this so difficult?*" while trying to make text fit correctly in a frame.	the user expresses surprise	User has to fuss with the positioning of frames as well as the size of the frames.
E3	4	0:26:47	User highlights a table of contents entry and scrolls down the Edit menu, but the Cut menu item is inactive. User also tries the delete button without success. User says, "*I can't seem to delete this entry. I don't know how to do it*."	the user explicitly gives up	The user tries several things to delete a TOC entry and then explicitly gives up.

Table 9. Two examples of problem reports. The first problem example was identified by evaluator E1 while analyzing the first user videotape. The second problem example was identified by evaluator E3 while analyzing the last user videotape.

After reporting the first results on this study in (Jacobsen et al., 1998a) we extended the study by investigating problem severity as a possible explanation for the evaluator effect. We therefore asked the four evaluators to judge the problem severity of all problems revealed. In order to delimit a bias towards judging those problems severe that the evaluator herself had initially detected, the problem list did not contain information on who had detected which problems. The usability problem master list consisted of

(1) a short description of each unique problem
(2) the number of users experiencing each unique problem
(3) the number of evaluators detecting it
(4) the usability criteria it was attributed to (one or more of the nine predefined criteria)
(5) the interface feature it involved.

In addition to the problem list, the evaluators received a scenario in which a project manager had constrained the evaluators to point out the ten most severe problems (hereafter called the top-10 list). The reason for constructing these top-10 lists was that a tight deadline forced the developing team to fix only the most important problems as the next release of the Builder was close at hand. In the scenario, the evaluators were told that their selection of severe problems should be based on the information on the problem master list and on other factors, such as considerations concerning experienced versus novice users, and the usage of Builder in real-life settings. The problems on the top-10 lists were not requested to be presented in a prioritized order, but each problem on the top-10 list should be annotated with the evaluator's reasons for including that particular problem on the list.

5.3.6 Procedure for the compilation of the analyses of the evaluators' responses

The four evaluators' reported a total of 276 usability problems. Based on a two-page structured coding scheme, two investigators (Morten Hertzum & I), created a master list of unique problems as follows. First, each investigator split up those original problem reports that he thought contained more than one unique problem. I split 16 original reports, producing an additional 23 problem reports; Morten split 17 original problem reports, producing an additional 18 reports. Eight of these new problem reports were the same. Hence, both authors had 284 problem reports in common (with Morten having an additional 10 problem reports, and I having an additional 15 problem reports).

Each investigator then examined their lists to eliminate duplicates. Of the 284 problem reports common to the evaluators the researchers agreed on 245 (86%[11]) as to whether they were unique or duplicated. This calculation was compiled as follows: if both investigators agreed that one problem report was not matched to any other problem reports, this counted as an agreement in the problem matching procedure. If one of the investigators but not the other investigator matched one problem report with at least one other problem report, this counted as a disagreement. Finally, if both investigators matched a problem report to the same group of other problem reports, this was counted as an agreement.

For those problems that the investigators disagreed on, they both made a specific independent proposition in order to reach consensus. There were no major disagreements in this consensus process of creating the master list as most initial disagreements were caused by differences in levels of abstraction.

The final master list comprised 93 unique problems (hereafter just called *problems*).

5.4 Summary of the results in the evaluator effect in usability test

In this section we will first describe the results of the user effect (section 5.4.1), then we will describe the results of the evaluator effect (in section 5.4.2), and lastly (in section 5.4.3) we will describe the results of the problem severity analysis as a possible explanatory cause for the evaluator effect.

5.4.1 The user effect

From previous studies on the user effect in usability test we would expect to see an increased number of usability problems for each additional user investigated in our usability test. And so we did (see Figure 11). On average one of our evaluators detected 19 problems analyzing one user, 31 problems analyzing two users, 39 problems analyzing three users and 48 problems analyzing all four users. Hence, our study also showed a diminishing return for each additional user analyzed if this is calculated by the formula ($\frac{\text{number of problems detected by n+1 users}}{\text{number of problems detected by n users}} - 1$). With this calculation an additional 63% problems are detected going from one to two users, an additional 26% problems are detected going from two to three users, and an additional 23% problems are detected going from three to four users[12]. In absolute numbers, however, the diminishing return is less obvious, being 12, 8, and 9 respectively. This less convincing diminishing return in absolute numbers for additional users might be an effect of too few users in our study. Hence, with more users we would most likely have seen an asymptotic curve or at least a curve only increasing slowly for each additional user analyzed.

[11] In the first paper, (Jacobsen et al. 1998a), we mistakenly reported the inter rate reliability to 84%; this was corrected in the second paper on the evaluator effect in usability studies (Jacobsen et al., 1998b)

[12] These percentages cannot be compared to the percentages in the published papers as our calculation here is based on one evaluator (averaging over the four evaluators in the study), while the numbers in the papers are based on all four evaluators.

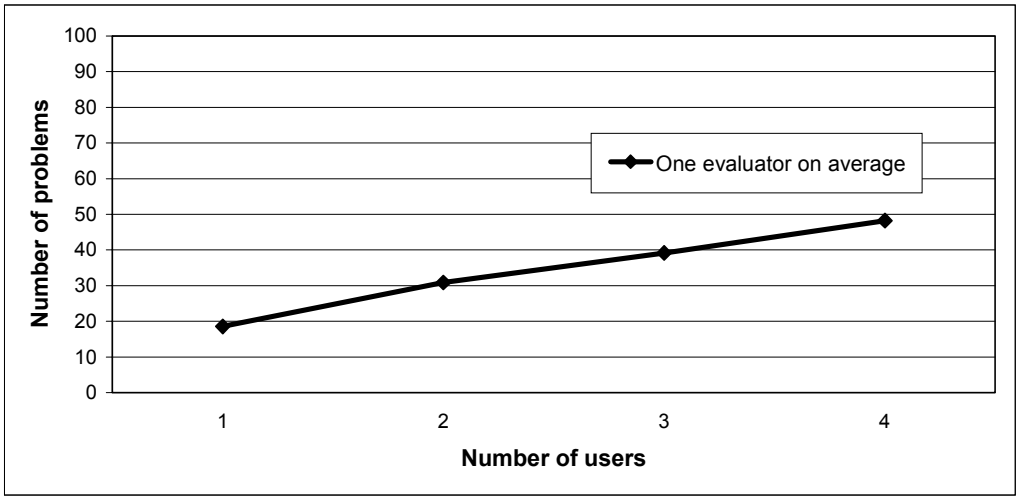

Figure 11. Number of problems revealed by one evaluator on average for one, two, three, and four users again on average. Analyzing an additional user increases the overall number of problems revealed. However, as expected there is a diminishing return for each additional user analyzed with an additional return of 63% going from one to two users, 26% going from two to three users, and 23% going from three to four users.

5.4.2 The evaluator effect

E1, E2, E3, and E4 detected 63%, 39%, 52%, and 54% of the total of 93 problems respectively. Thus, a single evaluator detected on average 52% of all known problems in the Builder interface. Figure 12 shows the agreement on problem detection for the four evaluators with the rows being evaluators (in the order E1, E2, E3, and E4 from top to bottom), the columns being all known usability problems, and the black squares representing a problem detected by a given evaluator.

Figure 12. Matrix showing who found which problems. Each row represents an evaluator, each column a problem, and each black square that the evaluator detected the problem.

The evaluator effect resembles the user effect (see Figure 13). On average one evaluator detected 19 problems analyzing one user, two evaluators detected 27 problems, three evaluators detected 34 problems, and all four evaluators detected 39 problems analyzing the same user (calculated as an average of all four users). Hence, our study also showed a diminishing return for each additional evaluator analyzing the same user with an additional 42% problems detected going from one to two evaluators, 26% going from two to three evaluators, and 15% going from three to four evaluators[13]. With more evaluators we would most likely have seen an asymptotic curve or at least a curve only increasing slowly for each additional evaluator analyzing the same user.

In Figure 14 we have added another three sets of data points to the ones depicted in Figure 11, i.e., the lower most set of data points in Figure 14 is identical to the one in Figure 11. The three additional sets of data points in Figure 14 represent number of problems detected by four, three, and two evaluators going from top to bottom. Hence, the effect of adding more users to a usability test with

[13] These percentages cannot be compared to the percentages in the published papers as our calculation here is based on one user (averaging over the four users in the study), while the numbers in the papers are based on all four users.

four evaluators is shown as the top most set of data points, while the effect of adding more evaluators to a usability test with four users is shown as the right most data points.

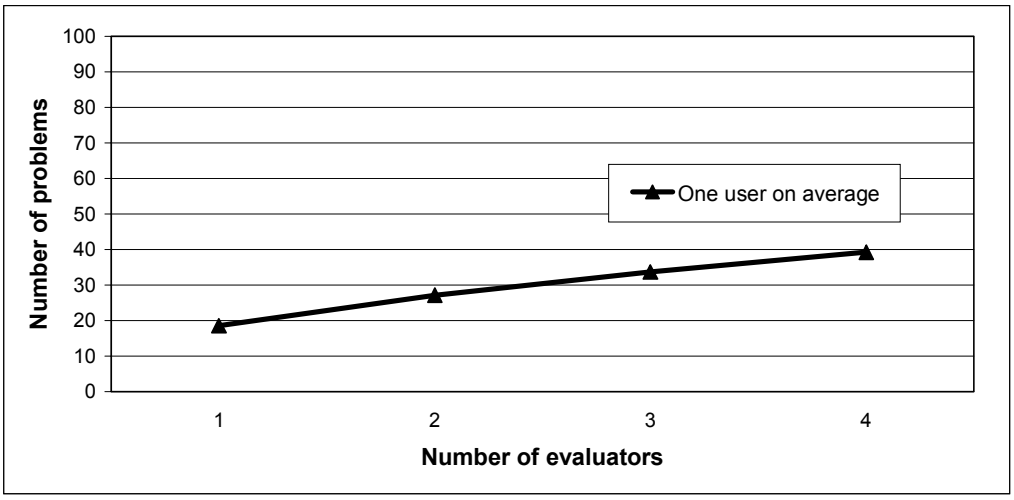

Figure 13. Number of problems revealed by one, two, three, and four evaluators on average when analyzing one user again on average. Having an additional evaluator analyze the same user increases the overall number of problems revealed. As for the user effect, there is a diminishing return for each additional evaluator analyzing the same user with an additional return of 42% going from one to two evaluators, 26% going from two to three evaluators, and 15% going from three to four evaluators.

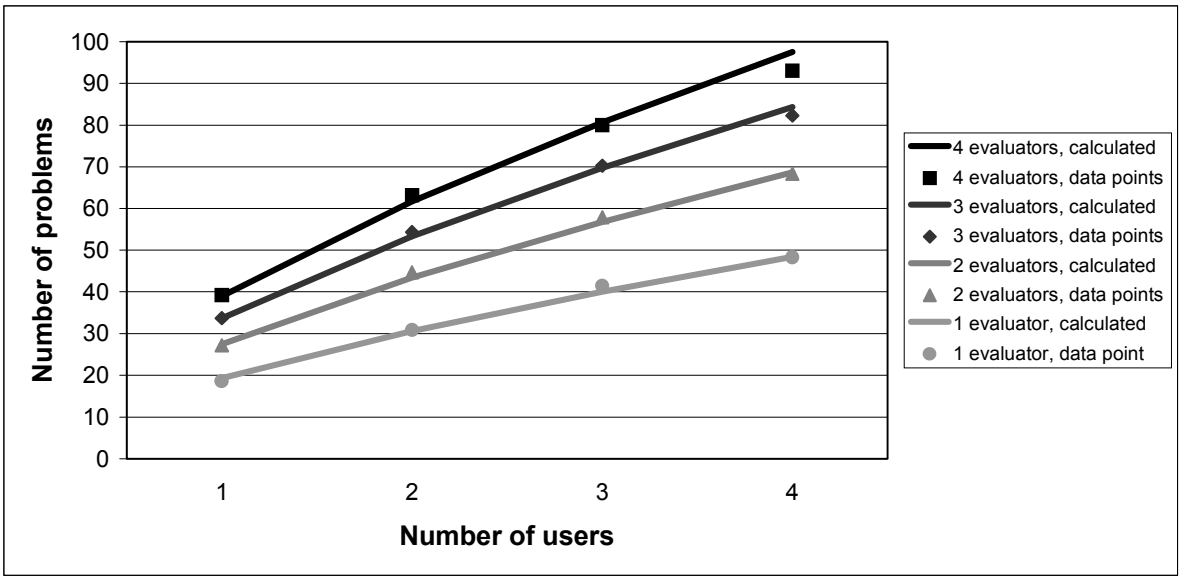

Figure 14. The number of detected problems depends on the number of users and the number of evaluators. Data points represent the results observed in the evaluator effect study, while curves represent results calculated using the equation *Number of problems = 19.35 * (number of evaluators)$^{0.505}$ * (number of users)$^{0.661}$*. The lower most data points show how many problems one evaluator detects on average for one, two, three, and four users (the exact same points are shown in Figure 11). The upper most data points show how many problems all four evaluators detected for one, two, three, and four users. The left most data points show how many problems one, two, three, and four evaluators would detect when analyzing one user on average (the exact same points are shown in Figure 13).

Our data can be described with equation 1. The equation describes the relationship between the number of detected problems, the number of evaluators, and the number of users in our study. The equation was found by running a linear multi-variant regression analysis on the data points in Fig-

ure 14 after a logarithmic function had been applied to the data points. The squared correlation coefficient is high ($R^2 = 0.997$; standard error of estimate =2,6%)[14]. Other studies may result in different values for the constant and the exponents. The number of problems calculated from the equation is also showed as curves in Figure 14.

$$\text{Number of problems} = 19.35 * (\text{number of evaluators})^{0.505} * (\text{number of users})^{0.661}$$

Equation 1. The equation was found by running a linear multi-variant regression analysis on the data points in Figure 14 (after a logarithmic function had been applied to the data points). The squared correlation coefficient between the equation and our data points (R^2) = 0.997; standard error of estimate = 2,6%.

Equation 1 is quite close to a geometric mean, see Equation 2. A geometric mean might be a more appropriate and a more general rule of thumb for understanding the number of problems that can be expected to be identified in a usability test based on number of users and number of evaluators. The important point in Equation 2 is that the number of evaluators and the number of users should be distributed equally to reveal the most problems in an interface. For example, conducting a usability test with 7 users and 1 evaluator will reveal fewer problems than conducting the study with 4 users and 4 evaluators. Note, however, that evaluators analyzing usability test sessions through videotapes require more resources than users participating in usability tests. Hence, although the geometric mean might apply in theory, resource considerations might imply that practitioners choose to include more users than evaluators in real-life settings. More studies have to be conducted to know if it is a general rule that the exponent part for users in Equation 1 is higher than the exponent part for evaluators, or if those exponent parts varies with different systems, users, evaluators, and settings.

$$\text{Number of problems} = \text{constant} * \sqrt{\text{number of evaluators} * \text{number of users}}$$

Equation 2. A more general and easy to remember equation describing number of problems as a geometric mean of number of evaluators and number of users.

5.4.3 Problem severity as a possible explanatory factor for the evaluator effect

The evaluator effect in our study was substantial. As much as 46% of the problems were not reported by more than one evaluator, while only 20% of the problems were reported by all evaluators. We hypothesized that if the severe problems to a larger extent were identified by all evaluators, while the differences among evaluators were caused by identification of cosmetic problems, the evaluator effect would not be as critical in real development projects. Hence, we employed three methods to extract severe problems from the total of 93 problems. First, we extracted the problems attributed, by any evaluator, to one or more of the three usability criteria (see section 5.3.4) that we considered more severe than the rest, (1) the user articulates a goal and cannot succeed in attaining it within three minutes, (2) the user explicitly gives up, and (8) the system crashes. We extracted 37 problems using this first method. Second, we extracted all problems that existed on at least one of the four evaluators' top-10 lists. We extracted 25 problems using this second method. Third, we

[14] The significance level (standard error for p estimate), which has been reported in appendices 4 and 5, does not apply in this setup, as it later became apparent to us that the data points are not independent of each other. Averaging over all possible combinations exactly re-use data points and hence, violates the assumption that data points should be independent of each other.

extracted those problems that existed on more than one of the four top-10 lists. We extracted 11 problems using this third method (see Table 10).

Detected by exactly	All problems	Violating criteria 1, 2, and 8	Any problem on the four top-10 lists	Problems on at least two top-10 lists
1 evaluator	43 (46%)	8 (22%)	5 (20%)	1 (9%)
2 evaluators	19 (20%)	7 (19%)	5 (20%)	3 (27%)
3 evaluators	12 (13%)	7 (19%)	2 (8%)	0 (0%)
4 evaluators	19 (20%)	15 (41%)	13 (52%)	7 (64%)
Number of problems	93 (100%)	37 (100%)	25 (100%)	11 (100%)

Table 10. The table presents the number of problems detected by exactly one, two, three, and four evaluator(s) for four groups of problems: all problems, those violating problem criteria 1, 2, and 8, those found on the four top-10 lists, and those found on more than one of the top-10 lists. The percentages of problems detected by only one evaluator decreased for each smaller group of presumably severe problems (upper row). The percentages of problems detected by all four evaluators increased for each smaller group of presumably severe problems (second row from bottom).

At first blush our hypothesis seemed to be confirmed. The percentages of problems detected by all evaluators increased for each smaller group of presumably more severe problems. Similarly, the percentages of problems detected only by one evaluator decreased for each smaller group of presumably more severe problems. Hence, severe problems had a higher probability of being detected compared to less severe problems. From this result we might further hypothesize that the evaluator effect primarily seems to be explained by problem severity, and secondly is less critical in practical usability tests, since usability staff are more interested in the detection of severe rather than cosmetic problems. However, as we will discuss further in the next section, there are also obstacles that might make this hypothesis less attractive.

5.5 Discussion

In the first sub-section of this section we will discuss previous methodologies for extracting severe problems. The second sub-section discusses the extraction methods employed in our study. In the third sub-section we will reveal supporting evidence for the evaluator effect by discussing the results of a paper by Molich et al. (1998), who simultaneously conducted an independent study closely related to ours.

5.5.1 Previous methodologies for extracting severe problems

In practical usability work, the process of extracting severe problems from insignificant problems is critical, since there might not be resources to correct all usability problems identified. Thus, the severe problems should be extracted from the cosmetic problems so that the severe problems can be repaired rather than the insignificant problems.

When investigating usability tests and other UEMs in research settings, severe problems have been extracted in order to qualify results of UEMs. For example, researchers wanted to reveal if one UEM faired better than another with respect to identifying severe problems. We have found a number of studies extracting severe problems in different ways. In one study, Connell & Hammond (1999) compared two UEMs to each other (usability principles and heuristic evaluation), and asked subjects to judge severity of each identified problem without further defining what constituted a severe problem. Desurvire et al. (1992) and Desurvire (1994) compared the outcomes of CW and heuristic evaluation with usability test and asked the usability test evaluator to rate problems according to a three-point problem severity code; 1 = minor annoyance, 2 = problem caused error, and

3 = caused task failure. In Jeffries et al. (1991), seven individuals rated the severity of problems identified using four UEMs: usability test, CW, heuristic evaluation, and guidelines. Raters were told to take into account the impact of the problems, the frequency with which it would be encountered, and the relative number of users that would be affected. Karat et al. (1992) compared empirical tests to individual and team walkthroughs and let evaluators rate problem severity according to a two-dimensional scale. One axis represented the impact of a usability problem on user's ability to complete a task and the other represented frequency, i.e., the percentages of the users, which experienced a problem. Molich (1994) defined something to be a serious problem if the user is unable to proceed without help from a human being or the user experiences annoying, irrational behavior from the system. He then continued by defining a user interface disaster as if, during a usability test, at least two motivated and typical users of the system run into a serious problem. Nielsen (1992) and Nielsen (1994) defined major usability problems as those that have serious potential for confusing users or causing them to use the system erroneously. Virzi (1990, 1992) used a seven-point scale and Virzi (1993) used a three-point scale to judge problem severity in order to test whether problem detection and problem severity were correlated. Virzi did not further describe what constituted a severe problem.

It is a general picture that extraction of severe problems is a common activity when investigating UEMs. Researchers use various scales and definitions as their basis for extracting severe problems. However, all studies are heavily based on an evaluator's judgment as the prime method for extracting severe problems; the severity definitions, if they exist, are so underspecified that they do not help evaluators in the process of reliably extract severe problems. We followed this tradition in our study, although one of our extraction methods was based on usability criteria. First, we chose a definition of problem severity based on the seemingly most serious usability criteria used by the evaluators (system crash, users give up, and users working for three minutes without progress). Then we asked the evaluators to extract the ten most severe problems based on a scenario where a tight deadline forced the developing team to only fix the most severe problems before system release. In this extraction procedure the evaluators should consider various aspects of usability, as for example, user frequency, evaluator frequency, novice and expert usage of the software, etc.

5.5.2 Discussion on methods to judge problem severity in our study

In order to get a deeper understanding for the results in our study, we further analyzed the data and found four obstacles that might dismiss the hypothesis that the evaluator effect is partly caused by disagreements on cosmetic, rather than severe, problems. First, there were disagreements among evaluators in their use of usability criteria for identical videotape sequences. Second, using breakdown situations[15] as the only measure for extracting severe problems might be wrong. Third, the evaluators' top-10 lists differed greatly when measured quantitatively. And fourth, concurrent and retrospective data indicated qualitatively different basis for constructing their top-10 lists. In this section we will scrutinize each of these obstacles.

One method to extract severe problems was to pick those problems that were attributed, by any evaluator, to usability criteria number 1 (no progress within three minutes), number 2 (user gives up), and number 8 (system crash). The agreement in the usage of the usability criteria for the 37 problems extracted in this group of severe problems was disappointing. Of the 37 problems 17

[15] According to Preece et al. (1994, p. 71), a breakdown situation is used to describe various incidents, problems, inefficiencies, mishaps, and accidents that arise in a work situation. By using our nine usability criteria we hoped to cover all sorts of breakdown situations in a usability test.

(46%) were attributed to different usability criteria for the exact same breakdown situation[16]. An additional 8 problems (22%) were only detected by one evaluator, and four problems (11%) were detected by different evaluators but not for the exact same breakdown situation; hence no disagreement could possibly occur for these (22+11 =) 33%. Only 8 problems (22%) of the 37 extracted problems were attributed to the same criteria for the same breakdown situation for several (but not necessarily all) evaluators. These figures indicate that we cannot rely on evaluators' use of usability criteria as a mean for extracting severe problems, simply because attributing a usability criteria to a breakdown situation is unreliable. Thus, several of the 37 problems extracted by use of this method were subjective in nature and might therefore not be considered severe.

One other concern regarding using breakdown situations as a method for extracting severe problems is that the three criteria used in this study exactly represent the criteria that are possibly the easiest for the evaluators to apply to breakdown situations. Intuitively, it would be hard to miss a system crash simply because the facilitator then would be forced to help the user start over again. Similarly, if the user explicitly gives up, the evaluator should notice this. Since all evaluators used a SDA-tool with a built-in stop-watch in the analysis process, it should not be hard to use the last usability criteria for extracting severe problems, namely the three minute without progress criterion. One the contrary, a user suggestion could be phrased in a way that triggered some evaluators to report a problem and other evaluators to judge the verbal report as not being a user suggestion. Similarly, the way of counting "three tries" was not further specified, so some evaluators might count on mouse clicks and others on the appearance of an executed function. Hence, one could argue – as we did in section 5.4.2 – that the three criteria capture severe problems, and that severe problems are seemingly less affected by an evaluator effect than all problems detected in the study. But, another causal analysis could be that the three severity usability criteria are easier to apply to breakdown situations than the remaining six criteria, and therefore the three criteria are naturally less affected by differences among evaluators. Without a method for identifying severe problems reliably, which is independent of the breakdown situation, we will not know whether problem severity and problem detection is associated.

We asked each evaluator to create a so-called top-10 list with the most severe problems. The detailed scenario for selecting severe problems was communicated to all evaluators and it resembled a realistic situation. However, a quantitative analysis revealed that the evaluators differed substantially not only with regard to the problem detection but also in the severity judgment situation. A total of 14 out of the 25 problems (56%) appearing on any top-10 list were only selected by one evaluator; 7 problems (28%) were selected by two evaluators, and 4 problems (16%) were selected by three evaluators. Not a single problem appeared on all four top-10 lists. It should be noted that the master list that was the basis for the top-10 list creations did not reveal which problems were initially detected by who. Despite this blinding procedure, one might think that the evaluators would be able to remember which problems they initially detected and therefore preferred to add these to their own top-10 list. This potential bias, however, did not seem to occur in our study. The data showed that the number of problems initially detected by each evaluator matched the proportion of problems on their top-10 lists.

The quantitative results that revealed substantial disagreements in the contents of the evaluators' top-10 lists are supported by concurrent and retrospective data collected in and after the selection of severe problems. The evaluators were asked to give reasons for adding a problem to their top-10

[16] A breakdown situation is considered the same when at least two evaluators report a problem for the same user pointed to the same time on the video tape (± 2 minutes).

lists (i.e., concurrently). After we discovered the disagreements in the evaluators' top-10 lists, we – retrospectively – asked the evaluators to describe their procedure in the selection of severe problems. Although caution should be exercised when using retrospective data, the results of the retrospective analysis were well in line with the concurrent data and the quantitative results. The evaluators' selection methods were based on different strategies such as the evaluators' favor for certain user groups (one evaluator favored novice uses, while another evaluator favored expert users), the number of users and evaluators encountering a problem, the problem criteria violated, expectations about real-world usage of the Builder, etc. No two evaluators used the exact same procedure or had the exact same motivation to select severe problems, and some evaluators were far from each other in focus and judgment procedure. Despite this disagreement in their process, all evaluators had sound reasons to use their procedures as they did, seen from their particular perspective. Hence, selecting severe problems based on a real-life scenario is to a large extent a subjective judgment.

Problem severity can be judged by the evaluators who initially detected problems in the interface or by a different group of people not affected by the process of detecting problems prior to their severity judgment. Evaluators who initially detected problems in the interface will be fully able to understand the problem descriptions and the interface as they have worked closely with the interface and the videotapes prior to their severity judgment. However, such evaluators might be (but as shown in our study, are not necessarily) biased toward the problems they originally detected. Jeffries (1994) found that problem reports are often unclear and ambiguous. Hence, relying on severity judgments made by an external panel, who has not been involved in problem detection procedure, introduces uncertainty regarding the interpretation of the problem reports. In collecting severity judgments, one has to balance the risk of biased severity judgments from evaluators against that of misinterpreted problem reports from an external panel.

In summary, we have learned several lessons from further examining our initial hypothesis on problem severity being a major cause of the evaluator effect. We were aware of the lack of a standardized method to sort the 93 problems into two groups of severe and cosmetic problems. Therefore, we introduced three different methods to extract severe problems, one relying on breakdown situations, and two methods relying on the evaluators' judgment based on a real-life scenario. At first blush, the results of our severity analysis confirmed our hypothesis. Although the evaluator effect did not disappear even with the smallest group of severe problems, there was a clear tendency that problem severity affected the degree of the evaluator effect. That is, for smaller groups of presumably severe problems there were a steady decrease in the number of problems only detected by one evaluator and a steady increase in the number of problems detected by all evaluators. Hence, the evaluator effect seemed to be less problematic in real-life settings, since evaluators to a greater extent agreed in their detection of severe problems than cosmetic problems.

However, further analyses revealed fundamental weaknesses in the methods for extracting severe problems. Breakdown situations, as mean for extracting severe problems, caused two obstacles. First, evaluators did not agree on the usage of the usability criteria, which was the basis for extracting severe problems; hence, selecting severe problems through usability criteria is probably an unreliable method. Second, the three criteria used in this study also represented those criteria that seemed the easiest to apply to breakdown situations compared to the remaining 6 criteria. As we still have no empirical evidence that the three criteria actually capture severe problems, the positive correlation for this group of "severe" problems might be attributed to the ease of usage of the criteria rather than the severity of problems. Another concern was that the top-10 lists were quite different in contents partly because they were constructed from different foundations. Not a single problem existed on all top-10 lists.

A concluding remark on the severity analyses is that reliable extraction of severe problems from a set of detected problems by more evaluators is extremely difficult. We were aware of some of the obstacles before we set out to investigate problem severity as a cause for the evaluator effect. We learned a great deal from the study, and now after the study has been conducted we have still only seen the tip of the iceberg. These results also put previous studies extracting problem severity as an unproblematic activity into perspective (see those studies mentioned in section 5.5.1). We believe that further definition on what kind of severity one has to look for, is one way to seek for a more reliable method for extracting severe problems. Simply setting up a scenario for extracting problem severity or asking evaluators to judge problem severity on a x-point scale will not yield reliable severity extraction. It has to be defined whether problem severity should relate to novice or expert users, time to fix the problem, ease of learning or time to complete tasks, etc. More studies on extracting severe problems from a data set are needed to precisely understand how we might select and judge the severity of problems reliably.

5.5.3 Supporting evidence

Since the presentations of the evaluator effect studies in Los Angeles and Chicago in 1998, one paper has appeared that to some degree supports the results from our studies. Molich et al. (1998) presented a study comparing four professional usability labs (one Irish lab, one UK lab, and two US labs) conducting usability tests of the same system.

The purpose of the comparison between the labs was to reveal how these labs approached the task of carrying out a discount usability test on a product intended for new users. Molich et al. wanted to investigate the different approaches to professional testing that are being applied commercially today, to show strengths and weaknesses of each approach used, and to give input to an ongoing discussion about usability testing as an art or a mature discipline that turns out reproducible results. The link between the study by Molich et al (1998) and our studies (Jacobsen et al., 1998a; Jacobsen et al., 1998b) is that Molich et al.'s study represents a rather ecologically valid study, with little control investigating the different outcomes of different evaluators' usability test on the same product, but with different users and in different settings. On the contrary, our study is less ecologically valid, but with more control investigating the different outcomes of different evaluators' usability test, on the same product, with the same users, and in similar settings.

In Molich et al.'s study, the system (an electronic calendar system for MS Windows) was made available to the labs 3 to 4 weeks before a usability report was to be delivered to the moderator. The usability test scenario comprised 16 lines describing the company that created the calendar system, the intended user group for the application, the expected knowledge of the users, and the reasons why a usability test should be conducted, as well as suggestions to which parts of the application that should be tested. The labs were asked to carry out a "normal" usability test and report the results in a written report. The report also had to consist of deviations from their standard usability test, the resources used, and comments on how realistic the exercise had been.

The general result was that two teams (A and C) spent roughly 25 person-hours on the exercise, while two labs (B and D) spent three times more on the very same exercise[17]. This difference between the labs was caused by different time spent on analysis and reporting. One of the "quick" labs (A) tested 18 users, while the rest of the labs tested 4 or 5 users. Labs B, C, and D provided quantitative results in terms of specific problems in the interface. They reported 98, 25 and 35 problems

[17] In comparison the evaluators in our study spent 21 hours on average but they were not asked to write a report on the results.

respectively. Lab A did not identify specific problems. None of the teams provided problem severity judgments. The agreement between labs on problem instances was disappointing (see Table 11). No two labs agreed on more than 10% of the problems detected collectively by both labs[18], and in the worst case less than three percents of the problems were detected by both labs. Looking at the agreement among three labs, the figures naturally decreased further; only one problem was detected by all three labs (0.6%).

For comparison, we made the same type of analysis for the four evaluators in our evaluator effect study (see Table 12). We found that any two evaluators agreed on at least 37% of the problems collectively detected, and that no two evaluators agreed on more than 46% of the problems that they collectively detected. The two studies – comparing labs with little control on method and environment and comparing evaluators with a large extent of control – are certainly not directly comparable. The controlling factors in our study, we believe, caused the evaluator effect results to be rather moderate compared to more extreme differences among labs revealed in the study by Molich et al. (1998). We constrained the evaluators in our study to use certain predefined usability criteria; Molich et al. did not aim to control what constituted a usability problem. We constrained the evaluators to report problems in a similar way (reporting evidence, free form problem description report, and association to one of nine predefined usability criteria); Molich et al. did not aim to control reporting procedure. We constrained the evaluators to detect problems using the same videotapes of four users; Molich et al. did not aim to control number, and types of users, and indeed they had to be different for each lab in their study. With those methodological differences in the two studies it is no surprise that the evaluator effect in our studies were less extreme than what was revealed in the study by Molich et al.

	Lab B	Lab C	Lab D
Lab B	-	3 (2.5%)	8 (6.4%)
Lab C	3 (2.5%)	-	5 (9.6%)
Lab D	8 (6.4%)	5 (9.6%)	-

Table 11. The agreement in problem detection among any two of the three labs that presented problem instances in their reports evaluating the same system (Molich et al., 1998). No two labs agreed on more than 10 percentage of the problems detected by both labs. Since Lab A did not reveal specific problems, they are not included in the table.

	E1	E2	E3	E4
E1	-	28 (41.8%)	31 (40.8%)	32 (41.6%)
E2	28 (41.8%)	-	23 (37.7%)	27 (45.8%)
E3	31 (40.8%)	23 (37.7%)	-	28 (40.0%)
E4	32 (41.6%)	27 (45.8%)	28 (40.0%)	-

Table 12. The agreement in problem detection among any two of the four evaluators who evaluated the same system (Jacobsen et al., 1998a & 1998b). Any two evaluators agreed on more than 37% of the problems that they collectively detected, but no one agreed on more than 46%.

The results from the study by Molich et al. (1998) support our results. The evaluator effect does not seem to be a phenomenon only appearing in one study. In fact the effect of different evaluators conducting a study on the same system in real-life settings is far more critical than what we found in a controlled environment. Our study aimed to reveal the "pure" evaluator effect holding several fac-

[18] The percents were calculated using the following formula: (number of problems in common by the two labs) divided by ((number of problems detected by lab *i*) plus (number of problems detected by lab *j*) minus (number of problems in common by the two labs)).

tors constant among evaluators, while Molich et al. went into the industry and tested the differences among labs across countries and communities. In order to meet the demands of scientific rigor, the methodology in our study may be stricter than what is applied in real-life settings. Molich et al.'s study has, on the other hand, a high degree of ecological validity in favor of scientific rigor. Both studies revealed similar results: the evaluator effect is substantial in usability test.

5.6 Conclusion

Our studies on the evaluator effect in usability test showed that analyzing usability test sessions through videotapes is an activity subject to considerable individual variability. When four research evaluators with extensive knowledge in HCI evaluated the same four usability test sessions, almost half of the problems were detected by only a single evaluator, while just 20% of the problems were detected by all evaluators.

Through different methods of extracting severe problems we found that the evaluators' detection rate was higher for more severe problems, but all sets of severe problems still displayed a substantial evaluator effect. However, further analysis on the extraction methods also revealed that none of them were sufficiently reliable to conclude whether real severe problems and problem identification are correlated or not. This result suggests that we need better methods and probably better definitions on what constitutes a severe usability problem.

The evaluator effect that is revealed in this study shows that usability tests are less reliable than previously reported. No single evaluator will detect all problems in an interface when analyzing usability test sessions, and any pair of evaluators is far from identifying the same set of severe problems.

A number of factors limit the generalization of the results in our study. The small number of users and evaluators participating is one limitation, another is that only one system was evaluated. Different evaluator experience and a minimum of training in analyzing the videotapes similarly among evaluators also limited the results to the way we chose to conduct the study. Finally two of the evaluators compiled the analyses of the evaluators' responses, which might have introduced unintended bias (Rosenthal, 1963; Tversky & Kahneman, 1974), although we deliberately blinded this factor in the compilation as much as possible.

5.7 Implications

Although our results are limited, we believe that the evaluator effect in usability test is a real phenomenon worth considering when using the method. Therefore, we risk our skin to suggest how the evaluator effect might implicate on the work of researchers and practitioners.

Usability test researchers are those who theorize, review, or conduct studies related to the usability test method. For those researchers who need to conduct usability tests, either to compare the results to other UEMs or to investigate some aspect of the method itself, the evaluator effect is a threat to the reliability of their results. In our study the evaluator effect resembled the user effect. The more evaluators analyzing the sessions the more problems were encountered. In fact, quantitatively, adding one additional evaluator to a usability test increased the total number of problems almost as much as adding one additional user. Hence, we suggest to add more evaluators to a research usability test and not only more users. If the formula of a geometric mean holds tight, there should be as many evaluators as users participating in a research usability test. This way, the most problems are identified with the least number of participants (users and evaluators).

Practitioners using usability tests have a different usage situation than researchers. While tight deadlines, limited resources, and a demand for quick communications of the results constrain practitioners in their daily work, researchers primarily search for scientific rigor results. Obviously, there is a subtle balance between rigorous studies and flawed studies. In the HCI community there is an on-going discussion whether some not rigorously conducted usability tests are better than no usability test. In my opinion there is definitely a limit to when the limited resources spent on usability work are wasted because the usability tests are conducted insufficiently. As the evaluator effect is probably more extreme in the industry than in experiments conducted in research settings, we believe that it is important to add more than one evaluator to a usability test. As the user is less expensive than the evaluator in respect to their time spent on each session[19], adding as many evaluators as users to a usability test in the industry might simply not be possible due to limited resources. However, two evaluators analyzing all users should be a minimum in industrial settings (in our study a second evaluator found 42% additional problems). In the future, the usability test should be developed to somehow decrease the evaluator effect, in favor of simply adding more evaluators to a usability test.

[19] Our users spent one hour each per usability test videotape, while our evaluators spent 3.5 hours on average analyzing one user on videotape.

6 Discussion

In this chapter we wish to discuss and put our work into perspective; we aim to dig deeper into the literature to get closer to questions like: why did we find such substantial evaluator effects? What have we learned from conducting case studies? What kinds of studies do we need in the future?

The first section discusses plausible reasons for the substantial evaluator effect in CW and usability test. The next section reflects on our case study work. We wish to describe why case studies are an important tool in understanding aspects of UEMs, and we suggest future work based on our experiences. Finally, we will put the current tradition of investigating UEMs into perspective. We describe why this tradition (which we have followed) is limited in its scope, and discuss what might be studied in the future and how this should be studied. We end this chapter and close this thesis summary with an epilogue.

6.1 Three plausible causes for the evaluator effect

A common theme throughout our studies has been the differences among evaluators, whether we investigated the evaluator effect in CW and usability test, the user effect in usability test or qualitative aspects of different evaluators using CW. Why did we find these individual differences? Dependent on one's perspective, one can claim that what we found in our studies was caused by the nature of human beings – that human are essentially different, and perhaps more different than what we usually assume us to be. With this perspective, it is difficult to come up with a remedy that relieves us from differences among human – and hence also among evaluators. With another perspective, one can argue that the differences among evaluators were merely caused by UEMs being weak and unreliable with respect to usage and outcome of using the methods. Improving the UEMs could solve this problem. Finally, one might claim that our methodology in studying UEMs caused substantial individual differences in evaluators' outcomes. This criticism refuses any generalization of our results and suggests other ways to investigate the evaluator effect. In this section we will discuss these three perspectives. 1) Are our results an effect of a more general characteristic of human beings – that they are essentially different when acting in complex environments? 2) Are UEMs weak and unreliable? 3) Did the methodology employed in our studies cause an evaluator effect?

6.1.1 Individual differences as a reflection of the nature of human beings

From a psychological perspective the issue of individual differences is no new research area. Differential psychology (or correlational psychology) is as old as psychology itself. Measuring different peoples' abilities physically and mentally goes back to the first psychologists like Wundt in the late 1800s. Differential psychology is a field that has largely evolved separately from experimental psychology (Thorndike, 1954), probably because what experimenters either try to remove from their studies or what they find as error variance, is the matter of interest for differential psychologists. Differential psychologists originally intended to identify certain mental abilities through use of multivariate or factor-analytic methods used on large populations. On the contrary, under the assumption of homogeneity in subject's behavior (or through necessary selection of subjects with similar backgrounds) experimental psychologists traditionally use relatively small sample sizes in the search for changes in dependent variables caused by manipulations on independent variables (Dillon & Watson, 1996). The HCI community, according to Dillon and Watson, has mostly adopted the experimental psychologist's rather than the differential psychologist's approach. Two exceptions are Egan's (1988) studies on individual differences among users and Landauer's (1987,

1997) persistent attempts to make it clear for HCI researchers and practitioners that the degree of individual differences in HCI is greatly overlooked.

Egan (1988) studied users of different systems and found some surprising results. For text editing tasks, the differences in completion time among users were reported to be up to 7:1, for information search up to 10:1, and in programming tasks up to 50:1. Egan aimed to reveal the differences among users in order to take this into account when developing user interfaces. He proposed (ten years ago) that it was time to develop robust user interfaces using prototypes that could be linked to adaptive trainer systems and automated mastery learning. Some of these ideas have found their way into real-life development organizations. But essentially Egan's results – that there are substantial individual differences among users – are still valid and relevant.

Indexing consistency is a related discipline to HCI and is particularly interesting with respect to the evaluator effect. The well-known indexing consistency problem is that any two indexers, when indexing one and the same document individually, will select sets of indexing terms that are most likely not identical (Zunde & Dexter, 1969). Some of these indexing consistency studies begin in established libraries by looking for identical items (e.g., articles) that are mistakenly indexed twice. These items are generally found to be indexed differently with respect to indexing terms. In half of ten studies, which Zunde & Dexter (1969) describe, there has been found indexing consistency below 50% between any two indexers. And more of the studies described there had been found indexing consistency down to 13% between any two indexers.

Although it is evident that there exists such a phenomenon as individual differences, many researchers will argue that research does not achieve any body of knowledge by explaining a given phenomenon by individual differences. Other researchers will argue that when applying research results to real-life settings, individual differences might be much more prevalent than the phenomenon initially studied. Thus, on the negative side individual differences as explanation for some experimental results do not guide researchers in any specific direction to better understand what constitutes these individual differences. On the positive side, individual differences might exactly explain the many types of complex behaviors by human beings that cannot be explained better by extracting some more specific parameters as the cause of their behavior. The debate also points out a practical consideration. Even if parameters could be extracted from behavioral differences among individuals, they might be impossible to use as independent variables due to the parameter's vague definitions. These parameters include *motivation*, *expertise*, *cognitive abilities*, *skill acquisition*, *skill transferal*, *personality* and *cognitive styles*, and possibly also other factors. In our studies on the evaluator effect, all these parameters might partly explain the differences among evaluators.

In complex problem solving, *motivation* is determined by the subject's attitude towards the problem, the degree of goal specificity, and the reward system associated with this (Dehaemer, 1991). In our usability evaluator effect study, the reward system was a part of the competitive nature of the study: all evaluators knew that other evaluators analyzed the same videotapes, and hence we assumed that the evaluators would do their best to identify all problems in the interface. In the CW evaluator effect study and the case study the students (subjects) were graded on their process and product, which is a typical motivational factor to ensure that students perform at their best. In all our studies, we tried to minimize the effect of different levels of goal specificity as well as the subject's attitude towards the problem they were to solve by instructing the subjects clearly and similarly. Motivation, however, is not a matter of either/or. Most likely, some subjects were more motivated than others in our studies. It is unknown to us if such differences in motivation can explain or partly explain the results of our studies, just like we generally lack a method ensuring calibrated motivation among subjects.

Although subject *expertise* in HCI studies is one of the factors that is often said to be controlled, it is essentially only partly under control. Expertise is a broad term and even when defined further with respect to domain of expertise, the domain is again typically a broad term. The expertise of interest in our studies regarded subject's expertise in the usage of a particular UEM. In our evaluator effect study in usability test, two evaluators had more expertise than two other evaluators. Here, we did not attempt to control the subjects' initial expertise, and though two of the evaluators had similar and greater experience than the two others with, none of the evaluators in the two groups were in practice similar with respect to expertise, since they had all analyzed different usability test sessions with different systems, different users, in different environments, etc., prior to our study. In the evaluator effect study in CW, none of the subjects had experience with CW prior to our study. They were – experimentally – a homogenous group with respect to expertise. However, we could not control the subjects' exact expertise in CW before they conducted their experimental cognitive walkthroughs. Some might have read and understood Wharton et al. (1994) better than others, and some might have grasped the use of the method better than others during the introduction to CW. In the evaluator effect study in usability test, we found a weak tendency toward more problem detection for evaluators with more expertise, but due to few evaluators in the study it could not be revealed if this tendency was significant or not. Taking a broader perspective on expertise, our subjects in all our studies obviously had different expertise in parent disciplines like HCI, computer science, and psychology. These bits of expertise probably had some impact on the differences among evaluators that we found in our studies.

Cognitive abilities have been measured since the early days of psychology. Charles Spearman's concept of general intelligence builds upon the idea that several cognitive abilities are closely correlated and that this can be attributed to an underlying common ability, the so-called *g*-factor. So, for example, if a person has verbal abilities above average there is a high probability that she also has good abilities in arithmetic, and grammar (Gleitman, 1995). In our studies we did not check for any differences among evaluators with respect to cognitive abilities. We accept that cognitive abilities have an effect on individual differences in general, but we do not know to what extent it influenced our studies. A similar argument is proposed for *skill acquisition* and *skill transfer* abilities. The evaluators in the usability test, for example, were experts in HCI, and hence they had an extensive (although different) knowledge about evaluation in general, and UEMs in particular. How they transferred their skills to the task of analyzing user videotapes is unknown to us.

Lastly, *personality* and *cognitive style* are other relevant aspects that also contribute to individual differences in HCI work. Though it is evident that individuals are different with respect to personality, it has been difficult to agree about a reliable classification of personality within psychology. Some argue that there are numerous distinct personality traits like "shy-bold", "cool-warm", "submissive-dominant", "practical-imaginative", etc., while others restrict personality to only a few distinct traits like "introvert-extrovert", "neuroticism stability-emotional stability" (Dillon & Watson, 1996). Similarly, we find different claims for the number and types of dimensions in what has been labeled cognitive style. These include for example "verbalize-visualize" and "holism-serialism". Personality and cognitive style, we believe, had an impact on evaluators' performance in our studies, and hence probably also contributed to the evaluator effect.

In summary, individual differences have been documented in HCI, in psychology and in fields completely different from those. Individual differences comprise a number of parameters, for example *motivation*, *expertise*, *cognitive abilities*, *skill acquisition*, *skill transferal*, *personality* and *cognitive style*. Those parameters might all contribute to individual differences in any experimental and realistic situation. Many of these are not possible to control experimentally, partly because they are ill-defined and partly because they cannot be controlled practically in an experiment. The initial

aim of our evaluator effect studies was not to come up with explanations for a possible evaluator effect, but merely to reveal whether an evaluator effect existed or not. With no control of dependent variables in our evaluator effect studies we would run the risk of being accused for flawed studies. We therefore attempted to control more of the variables than what is typically seen in both practice and research settings. Despite this effort, we found substantial evaluator effects in usability test and CW. In section 6.1.3 we will examine if we could have investigated the evaluator effect in a more controlled manner, and what consequences this would have had for the generalization of our results. The next section looks into the particular UEMs that we investigated in order to judge whether the evaluator effect was caused by weakly described and unreliable UEMs.

6.1.2 The evaluator effect as a consequence of weakly described and unreliable UEMs

UEMs aim at identifying weaknesses in the interaction between user and system enabling developers to improve the system in question. One quality aspect of UEMs is that they should be reliable with respect to the evaluators. If a UEM were completely reliable with respect to the evaluators, two different evaluators applying the same UEM to the same interface should identify the same set of problems. In a worst case scenario, a completely unreliable UEM would imply that two different evaluators applying the same UEM to the same interface would identify completely different sets of problems. Looking at the results of any pairs of evaluators in the evaluator effect in usability test, the sets of problems detected by only one of the evaluators are constantly larger (between 55-63%) than the set of problems detected by both evaluators (between 37-45%). Hence, for the usability test it seems that we are closer to a worst case scenario than to an ideal scenario. The CW evaluator effect study is even closer to the worst case scenario when investigated in controlled environments. Taking those problems that have been collectively detected by any two CW evaluators and splitting them up into two groups, 17% is detected by both analysts and the remaining 83% is detected by either one of the two evaluators when averaged over all 55 combinations of pairs included in the calculation. The circumstances in the case study on the two analysts A1 and A2 were less controlled than the evaluator effect study. Thus, as would be expected the reliability of CW were even more disappointing. In the case study only 6% of the problems were detected by both analysts, while the remaining 94% were detected by either A1 or A2 but not by both evaluators.

Are there any reasons to believe that the description of usability test and CW is a major reason for the poor reliability of the methods? We believe that the methods are indeed weakly described and that this is a partial cause of the evaluator effect. Let us look at the weaknesses of the methods one at a time.

As described in section 2 the usability test can be used in various situations and in many different ways. Different usability test setups obviously yield different results. We have investigated one particular way of conducting a usability test in a rather controlled environment. In order to minimize the evaluator effect we even attempted to control critical incidents by predefining nine usability criteria to be used by the evaluators in their analyses of the videotapes. This attempt probably reduced the evaluator effect somewhat, although the overall results showed that it clearly was not enough to remove it. One weak aspect of the usability test is that there are no clear definitions on critical incidents in general. The *Handbook of usability testing* by Rubin (1994) is one of the most comprehensive books on how to conduct usability tests. This book does not include descriptions of how an evaluator should transform an observation into a usability problem. Actually, we have not been able to find any literature suggesting precise criteria to be used in the analysis phase of the usability test. The usability criteria used in our study are therefore our best guesses of what might lead to a problem detection. The usability criteria could certainly be improved both with respect to

their contents and their form. More precise usability criteria would be one possible way to decrease the evaluator effect.

The third version of CW (Wharton et al., 1994) is weakly described in many respects. As has been pointed out earlier, the four questions are subject to considerable confusion (e.g., John & Packer, 1995; John & Mashyna, 1997). We have hypothesized that the description of the fictive user is an important part of CW. If the fictive user description is too weak, the evaluator seems to both stereotype the answers to the four questions as well as anchor the answers to the evaluator's own experience. A minor although still visible factor is that a task can be transferred to several correct action sequences and each action can be described in several level of abstractions. Those weaknesses all contribute to the evaluator effect, and again, more precise descriptions will probably decrease it.

Are there any negative consequences of describing the two UEMs better with the aim of decreasing the evaluator effect? There is at least one problem. Attempting to increase the reliability of UEMs does not necessarily make them any more valid. Consider the targets in Figure 15. If two riflemen used the same rifle to hit the left-hand target we would conclude that the rifle was precise but with a systematic error forcing the riflemen to constantly miss the center of the target. If the same riflemen used another rifle to hit the right-hand target we would conclude that the rifle was less precise, but that it did fairly well in hitting the center of the target. In analogy with this story, we could imagine that the riflemen were evaluators, the rifle UEMs, and the hits being usability problems. The left-hand target shows a reliable UEM that unfortunately is not as valid as one would hope for (i.e., the UEM does not capture the central usability problems). The right-hand target shows the outcome of another UEM that is less reliable than the former but more valid, as it identifies more central usability problems. The figure shows that making a UEM reliable is not the only important consideration. Making usability test and CW more precise in their description in hope of an increased reliability might make them less valid. Unfortunately, we do not know to what extent the usability test and the CW produce valid results today. We have only found them to be unreliable. Thus, the shots are spread around on the target, but we do not know whether they are hit randomly around the center of the target, or if they systematically hit the left of the target, or if they hit the target at all. Our reliability studies merely indicated that the usability test and CW is indeed unreliable, and the question pushed forward now is, how valid are the two UEMs?

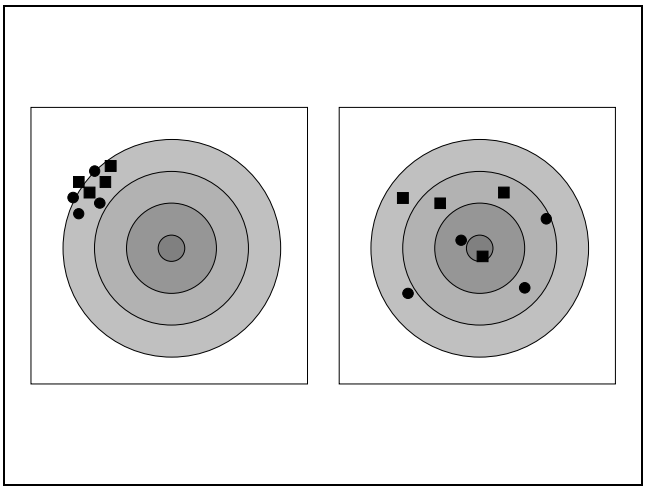

Figure 15. Imagine that two riflemen shoot four times each at the same target using the same rifle. One rifleman punches the target with round holes and the other with square holes. The target on the left side indicates that the rifle is a precision device, although it systematically misses the center of the target. The target on the right side indicates that the rifle is not as precise, but it hits fairly close to

the center of the target. If the rifle was a UEM, the riflemen evaluators and the holes in the target were usability problems, we would define the UEM on the left hand side as being reliable but with little validity, while the holes in the right hand target represent a UEM being less reliable but closer to identifying valid problems. (Inspired by Reason, 1990)

6.1.3 Did we use inappropriate methods for studying the evaluator effect?

After presenting the studies on the evaluator effect at conferences and meetings, we have discussed the results with other researchers and received various kinds of (often friendly) criticism to our methodologies. One obvious question is whether the evaluator effect is just an artifact of our setups? That is, if we had changed the study setup slightly, would we find better agreement among evaluators? We have scrutinized different criticisms to our studies both to document what could be done in future studies on the evaluator effect and to hypothesize if such methodological changes would have had any side effects.

In the usability test we have found three criticisms: 1) the recruited evaluators could have had more similar expertise level, 2) the evaluators could have received better training in using the usability criteria, and 3) we could have calibrated the evaluators knowledge about the system evaluated.

The recruited evaluators could have had more similar expertise level. Two of the evaluators in our study were quite experienced in analyzing usability sessions, and two others were less experienced. The more experienced evaluators detected slightly more problems than the less experienced evaluators. However, the agreement between the two experienced evaluators were no better than the agreement between one of the experienced and one of the less experienced evaluators. One way to control the evaluator experience would be to select evaluators from the same company with similar experiences in evaluating systems within that company. This would potentially minimize differences among evaluators' judgment of when they have identified a problem. However, we do not believe that such a change in the setup would threaten the overall findings in our studies. From practitioners, whom I have talked to, there seems to be an acceptance that different evaluators within the same company find different types of problems, e.g., in Kommunedata in Denmark (Zobbe, 1999) and in Nokia Mobile Phones (Hynninen, 1999). Usability evaluators in the industry are recruited from very different fields. They have received different forms of training and they focus on different aspects of the interaction between user and system. If we had the possibility to control evaluators' experience in an evaluator effect study, we would perhaps see better agreement among evaluators, but our data would also be less applicable to the universe we are trying to represent, namely the practical world of usability evaluators.

The evaluators could have received better training in using the usability criteria. Kirakowski (1998) raised this methodological criticism. Our usability test evaluators did not receive any training in using the nine predefined usability criteria. This criticism is perfectly valid; we did not give any formal training in the use of the usability criteria. We assumed that having usability criteria at all would decrease the evaluator effect compared to having no usability criteria. No doubt, having trained the evaluators in using the usability criteria would have even further decreased the evaluator effect. The trade-off was, again, whether we on one hand wanted to eliminate experimental factors that represented causes for resulting evaluator effect or we on the other hand wanted to design a study that could be generalized to practical usability tests. In the literature, we found no evidence that precise usability criteria are used when conducting usability tests (apart from performance measures in usability test). A problem is considered a problem when the evaluator judges something to be a problem. We thought this definition would indeed produce large individual differences among evaluators. Therefore, we introduced precise usability criteria in our study. Given that the usability criteria used in our study probably pulled in the direction of lesser rather than more differ-

ences among evaluators, we have showed that our results only partly represent the real world. In the real world – using no usability criteria – there would probably be greater differences among evaluators (see e.g., Molich et al., 1998). Hence, we have attempted to show that the evaluator effect indeed exists. In a future study it would be interesting to see if we could reduce the evaluator effect by setting up usability criteria and train the evaluators in using those criteria. As mentioned before, the criteria and the training in using them should perhaps succeed a study that investigates which usability criteria we should apply in order to achieve not only reliable, but also valid data.

A final criticism to our work on the evaluator effect in usability test is that we did not control the evaluators' knowledge of the system tested (the *Builder*). Rubin (1994) stresses that the evaluator's knowledge about the system tested is essential for the success of the evaluation. In our study two evaluators had a detailed knowledge about the Builder, while two others were asked to make themselves sufficiently familiar with the system so that they were able to analyze videotapes on usability test sessions. They spent 2 and 12 hours respectively making themselves familiar with the system. These numbers alone indicate that they did not agree about how much learning time would be sufficient to do the later analysis. There were, however, no direct indication that the evaluator spending only two hours missed problems because she lacked understanding of the system; both evaluators who had to make themselves familiar with the system detected an equal amount of problems in the Builder. There is no doubt that the evaluators' different knowledge about the system evaluated contributes at least somewhat to the evaluator effect. Not enough knowledge about the system evaluated makes the evaluator overlook some problems or describe the problems inaccurately. Extensive knowledge about the system perhaps introduces a bias towards finding evidence for describing a problem that the evaluator herself experienced in her learning process. Making oneself familiar with a system before analyzing usability test sessions is an individual activity. Allocating a predefined time to such an activity might make a study on the evaluator effect look more convincing but will not, we believe, remove the factor of learning time that is subject to individual differences. Moreover, the time spent in learning to use a system for an evaluator in real-life settings is by no means standardized; rather it varies greatly from system to system, from company to company and from situation to situation.

In the CW evaluator effect study we found two methodological issues that could be criticized. 1) The evaluators received an introduction by one of the authors of the paper (Hertzum & Jacobsen, 1999) and the evaluators' use of the CW was therefore partly limited to the quality of that introduction. 2) We did not monitor the evaluators during their work; hence they spent different amounts of time evaluating the system in question and they used different strategies for achieving supplemental knowledge about the use of the method from the practitioners' guide (Wharton et al., 1994).

Learning the use of a UEMs is important for the success of the evaluators' outcomes when using a UEM. Our evaluators received little training in how to use the CW before they applied the method to an interface. First they received a two-hour introduction in class settings and then the practitioners' guide to CW was handed out to them along with the task of evaluating an interface using CW. This methodology was employed because we wanted to investigate the evaluator effect for first time users of CW. The quality of the teaching was not controlled per se, although the teacher himself had received an introduction in using CW along with being well informed about the contents of the literature about CW. Any teaching situation can probably be improved. If the teaching had been conducted by CW developers, it would probably have looked different and have been of a higher quality than when an outsider (although professional in HCI) teach the evaluators. However, any UEM should be sufficiently robust to be taught by outsiders, and if the method does not allow such a distribution, it has no potential of being of any significance to HCI practitioners. In fact, we are aware of at least one case where reviewers have rejected a research paper because the trainer of the UEM

was also the developer of the UEM. The review committee agreed that this setup was biased insofar as that the positive results of the UEM were not general to the public because training the evaluators in the use of the UEM heavily affected the results of the study. With the information we have achieved by doing case studies on CW and the evaluator effect study on CW we would probably be able to increase the quality of the training of evaluators and, perhaps through this improvement, decrease the evaluator effect. However, it is more likely that the evaluator effect would continue to be of a size that does not permit us to believe that CW will become reliable. Our setup probably reflected the training most HCI students (and later evaluators) receive in a given UEM.

The other point of criticism of our CW evaluator effect study is that we did not monitor the evaluators' work process. The outcomes of the processes were our data material. These outcomes were of course different and had different levels of quality partly because the evaluators spent different amounts of time and different efforts in their attempts to achieve good results. We could have controlled several external factors in our study if the evaluators had been monitored. In future studies it would be valuable to test the evaluator effect while the evaluators are being monitored. However, we do not believe that the results would change significantly with this different setup. Also, our setup better represents how CW is conducted in real life settings, where evaluators are not monitored.

The results from our case studies on the usage of CW support the results from the evaluator effect study. In the latter study, evaluators received similar training, read the same paper and evaluated the same system with the same tasks. In the case study, we investigated two analysts with different learning scenarios that used different tasks evaluating the same system. Hence, the evaluator effect in a case study situation – resembling real-life settings – were much more evident than the evaluator effect was in more controlled and less ecologically valid settings. And still both studies displayed a substantial evaluator effect.

By scrutinizing various criticisms on our evaluator effect studies we found that the setup could be varied greatly. It is unclear how much different setups would impact the results of our studies. However, we do not believe that any setup changes would completely remove the evaluator effect in usability test and CW. A general trade-off in all experimental settings is whether one should seek for either high ecological validity combined with low degree of control or one should seek for high degree of control combined with low ecological validity. The criticisms we received on our studies could probably have been overcome by more control, e.g., by adjusting the evaluators' experience to each other or by training the evaluators in the use of the predefined usability criteria. However, increasing control results in pushing the studies further away from the situation that we initially wanted to investigate (UEMs used in practice), hence decreasing the degree of ecological validity. The general negative correlation between ecological validity and control is shown in Figure 16.

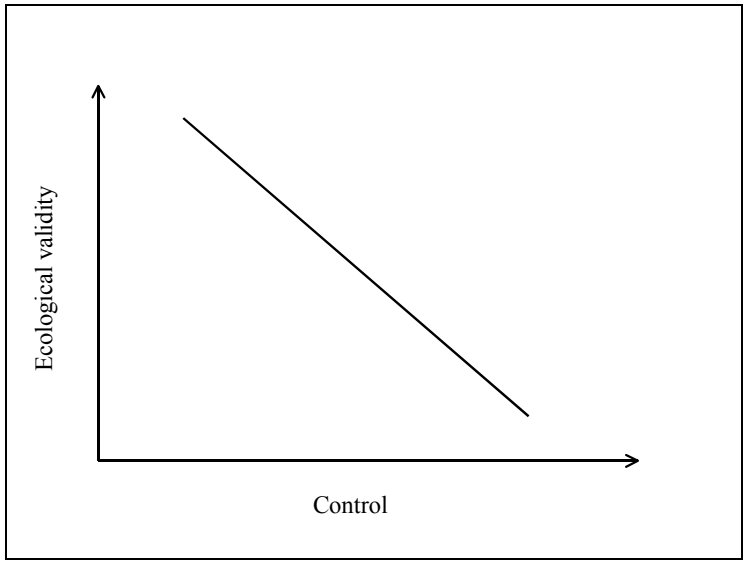

Figure 16. Ecological validity and control over factors in an experiment are negatively correlated to each other. With increased ecological validity one loses control over the factors in the experiment. On the contrary, by increasing the control in experiments, the degree of ecological validity decreases.

With these detailed and more general considerations on control of factors versus ecological validity, we will conclude that more studies are needed to test if our results can be replicated with slightly different setups, with different evaluators, different users, and different types of system. It is our belief that the evaluator effect is an effect that exists in usability work in research as well as in industry. The question to be raised next is what can be done to the evaluator effect, and how do we further investigate UEMs with regard to validity? (So far, only GOMS has been validated, Gray et al., 1993).

6.2 Reflections on using case studies for studying UEMs

Roughly half of my Ph.D. period has been spent on analyzing and writing about two analysts learning and using CW (Jacobsen & John, 1999; appendix 1). This approach is an unusual way of studying UEMs. HCI researchers, especially those having an experimental psychologist or a human factors background, tend to conduct experiments to answer UEM research questions. This might be part of the reason why UEMs are primarily studied quantitatively rather than qualitatively. There are reports on UEMs, though, that are qualitative. These reports are written by practitioners that often report case *stories* opposed to case *studies*. Case stories are not necessarily based on certain well-accepted methodologies, rather they describe a case situation, which inform others about the authors' cases, their experiences and opinions about some UEM. Case *studies* are equivalent to experiments and other scientific methods insofar as that they ask a specific theoretical question, which is investigated through stringent analysis of a carefully constructed experiment or case. This procedure allows for a theoretical (or in Yin's (1994) terminology: analytical) generalization. Case *stories*, as enlightening as they may be, do not have such theoretical underpinnings and, hence, do not enable analytical generalization of the results.

Conducting methodological rigorous case studies are just as difficult as conducting methodological rigorous experiments; and as Gray and Salzman (1998) demonstrated, conducting rigorous experiments is no easy task. We have followed Yin's (1994) premises and procedures for conducting case studies, not because case studies are more appealing than, for example, experiments, but because

the questions we wanted to ask about CW were of a kind that could not be answered by conducting experiments.

The aim of our case studies was to test one hypothesis and to further explore the learning process and usage of CW. The hypothesis we tested was; if one analyst was able to use CW with less confusion and better outcome when only reading Wharton et al. (1994), compared to another analyst reading several earlier papers on CW. We have been criticized that case studies cannot test hypotheses because of the few participants included in case studies. This criticism is unfounded. According to Yin (1994), case studies can actually test hypotheses. The analysis of case study material is based on multiple sources of evidence, establishment of chains of evidence, explanation building, and time series analysis. Thus, if multiple sources of evidence regarding a hypothesis point in the same direction, we are able to establish a conclusion of this specific hypothesis within the settings investigated. (Exactly this type of hypothesis can also be tested in an experiment, while exploration of contemporary phenomena within real-life context is reserved for case studies and other qualitative methods – but not experiments).

Another question is often asked, "How can you generalize from a single case?" The question, however, also applies to experiments, "How can you generalize from a single experiment?" The fact of the matter is that a single case study *or* a single experiment very seldom reveals scientific facts. Scientific facts are based on multiple experiments and multiple case studies. Also, both case studies *and* experiments generalize theories (analytic generalization), rather than enumerate frequencies (statistical generalization) (Yin, 1994, p. 10). Apart from using our case study to test one hypothesis we employed exploration of the case study material. We conducted two case studies with the aim of generating hypotheses (rather than conclusions) regarding processes related to learnability and usability of CW. These suggested hypotheses ought to be tested in future experiments and case studies.

Although the process of conducting case studies is regarded as time-consuming, it is not considered nearly as time-consuming as ethnographical and field studies (Yin, 1994). However, the highly iterative process of creating a database and entering in all relevant information into this database, not to mention the exploration of the database has, at least to me, proved that conducting case studies *are* time-consuming. Databases are convenient for structured data. Our data were quite structured, although not completely structured. Small drawings and footnotes made by the evaluators (A1 and A2) could be stored in our database, but they are difficult to retrieve and view. Hence, while databases are a convenient tool for structuring case study data, they also have their limitations.

We have put much effort into categorizing text data in order to easily retrieve relevant information from the database. This process has taught us a couple of lessons. Yin (1994) suggests that a database be created in order to separate data from interpretations and analyses of data. This separation of data and analyses of data appealed to us. In our classification process we discovered that by classifying data in the database, we had already begun analyzing the data. In fact, it became apparent that simply organizing data into database tables, and relations between tables, is akin to analyzing data. Going one step further back, it is clear that the methods and choices for collecting data in the first place partly determine the possibilities that one has in the analysis process. These problems in conducting case studies are perhaps not new to experienced researchers, but being new to the field I first really understood how data collection, data storage, and data analysis are closely related during and after conducting this study.

As we have experienced it, a complete separation between data storage and data analyses is not possible. However, it is a good principal rule to *try* to separate data storage and data analyses, as to enable other researchers to review data independent of the analyses. It is difficult for us to see how

one can possibly analyze a massive amount of data without some tool to structure the data and some support in retrieving parts of the data. Our study, at least, could not have been completed without storing data in a transparent data structure (see a detailed description of this in Jacobsen, 1999), from which exploration of data could be exercised.

Another lesson we learned in the analyses of our data is that when we wish to retrieve some specific type of data that the database has not previously been prepared for, we need to further classify data and, thus, change the contents of the database. The good side of this is that databases are extremely flexible regarding extensions of new type of classifications. Classifying data in relation to some theoretical proposition also makes it possible for us to reconstruct queries quickly and at any time. The negative side is that the classification takes time and may end up being fruitless – if the hypothesis either fails or if we, after hard work, discover that we cannot set up chains or collect multiple sources of evidence. In summary, databases are essential in case study work, they are reasonably effective, they retrieve reliable and valid data in seconds but they are also rather time-consuming to design, build, and prepare for exploration. We plan to make our database available to the public through the Internet, so that our effort can be shared with others interested in conducting case studies in UEMs.

One question remains. Was it worth the effort conducting a case study analyzing the learning and usage of CW? As we cannot generalize our results from the case study to a population but only to theoretical propositions, we need to see studies testing our hypotheses and suggestions to see whether our analyses faired in the right direction. Obviously, we believe that we found results that are valuable to CW practitioners and developers. And indeed results that have not been seen before, simply because UEM researchers have no tradition in conducting case studies. More case studies are needed with the aim of investigating CW and other UEMs to learn more about how they are learned and used.

6.3 The problem of defining a problem: the problem for the future

A recurrent problem in our work has been to understand what a usability problem is. The tradition of studying UEMs has been surprisingly stable the past 15 years. Hundreds of articles, papers, and conference contributions have viewed usability problems as something that is identified rather uncritically using some sort of UEM. In the parent discipline of Human Errors, researchers have tried to define and categorize human errors in various ways. One pragmatic classification is the tripartition of behavioral, contextual, and conceptual levels, corresponding to the "What?", "Where?", and "How?" questions about human errors (Reason, 1990). Another distinction is between error types corresponding to the cognitive stages of planning (an error being a mistake), storage (an error being a lapse), and execution (an error being a slip) (Rasmussen, 1983). For some reason there seems to be only little communication between the discipline of human errors and research into UEM. One exception is Cuomo & Bowen (1994) who compared the outcomes of different UEMs to each other and analyzed the results based on Norman's stages of user activity when using a system. The user stages are a cycle of cognitive activities modeling user's mental stages in the interaction between user and system (Norman & Draper (1986). However, Cuomo & Bowen (1994) or others in the human error discipline have neither given us any clear definition of what constitutes an error[20] nor what constitutes a usability problem. How should we report an observation in a usability test as a

[20] There are several classification systems of human errors, but I have not been able to identify rigorous criteria that enable observers to reliably identify human errors.

problem? What should we judge as a usability problem when using inspection methods? These questions seem to be truly important and yet have received no attention.

We believe that we need a better understanding of what constitutes a usability problem. In the literature in general, as well as in our studies, a usability problem detected in a usability test is treated as something definitive. The reported problem *is* as problem, simply because it is reported: "User X had problem Y". Our evaluator effect results in the usability test extended this understanding somewhat to: "User X had problem Y when analyzed by evaluator Z". It is not the user who experiences a problem. It is an evaluator who judges and reports a problem given an observation of a user. This point of view is more obvious when using usability inspection methods. Here, an evaluator judges something to be a problem. But, is a problem reported by an evaluator *really* a problem? Yes, in practice it *is* a problem, at least until it has been communicated to a developer who will perhaps deny that the reported problem is a real problem. Suddenly, the problem is an object that can be negotiated among agents (e.g., evaluators and developers). The problem is no longer definitive. In fact, the evaluator might get convinced that what he reported as a problem is actually no problem, rather it is a feature in the system (e.g., an intended obstacle in a children's game). In research settings, like ours, a usability problem is seldom treated as an object in a negotiation. Usability problems are definitive. Even a problem that is reported by a true mistake has the same value as any other problem, since evaluating the evaluators' reports would be regarded as introducing a bias rather than increasing the validity of the study.

The evaluator effect as a reliability issue is very closely related to the validity issue of defining a usability problem. And validity and reliability issues, which we want to ensure in science, get another meaning when applying the usability test in real-life settings. If a problem is an object in a negotiation between agents (e.g., the user, the evaluator, and the developer) one can possibly move the definition of a problem closer to any of the agents with the aim of increasing the reliability of problem identification. The users could, theoretically, be taught to use new systems in a way that enabled evaluators to detect problems more reliably. The evaluators could, also theoretically, be taught how to analyze users, and thereby to a greater extent agree on problem identification. But a more fruitful way would probably be to change the usability method in a way that enables evaluators to agree more on problem identification. Therefore, we believe that future studies should aim at changing both CW and the usability test method in order to increase their reliability. This might be done by further investigating the usage of usability criteria for the usability test and by developing a computer tool supporting a CW evaluation.

We do not believe, however, that there are definitive usability criteria. Usability criteria depend on the system under evaluation and the usage situation. A children's game should be evaluated with a different set of criteria than that of a multimedia learning environment, which again should be tested with yet another set of criteria than that of a cellular phone. Future studies should investigate which usability criteria that are suitable for which type of systems and situations. And perhaps there are more dimensions to usability criteria than just the system and usage situations.

Another issue for future studies is testing the validity of usability test and CW. This is, naturally, closely related to the previous discussion about what constitutes a problem. One thing is to judge what constitutes a problem in a usability test, another is to match those problems with real users' problems in real-life settings. Who should judge what constitute real usability problems to users and how should this be done? This question is definitely not easy to answer, although it is extremely relevant to the HCI community.

Finally, it could be valuable to describe and analyze the social interaction between agents in a usability evaluation situation as to get closer to the negotiations in what constitutes a problem in real-life settings.

6.4 Epilogue

In the eighties several UEMs were developed and presented at conferences, in journals and in textbooks. In the early nineties there was a great interest in comparing these UEMs to each other, refining them, and there were attempts to transfer them into the industry. The usability test is far from being used in all major development projects, but compared to other UEMs available, it is the most widely used UEM in the industry, and definitely the UEM with the best credibility in the HCI research community. The usability test is considered as the golden standard. CW, on the other hand, is a UEM that has received great attention in the HCI research community, but without being used much in the industry. It was found reasonably useful but also tedious and difficult to use as well as immature compared to other UEMs. Gray & Salzman (1998) found severe flaws in many of the important comparative studies of UEMs, and concluded in their review that we do not know as much about UEMs that we thought we knew.

We investigated CW qualitatively to better understand how CW was learned and used, initially with the hope of finding aspects that could be improved. We did find several problems with CW. With support from our quantitative study on the evaluator effect in CW we revealed that the outcomes of CW differed greatly with different evaluators. Thus, CW is not as reliable as we thought. Our case study also revealed that evaluators miss a great deal of problems when compared to problems observed in usability tests. Based on these results and detailed, qualitative process data, we suggested several changes to CW, with the hope of improving its accuracy and reliability.

In the process of using usability test to reveal the predictive power of CW, we began suspecting that usability test itself was perhaps not as reliable as we initially believed. We set out to investigate the evaluator effect in usability test and found surprisingly and huge differences between evaluators' problems lists when they analyzed the same videotapes. This work hopefully impacts the use of usability test in the industry and should also change our understanding of the usability test as a "golden standard".

As a reflection on our work, which is based on traditions in investigating UEMs, we have found it particularly interesting to ask the following research question: how do we understand, treat, and define a usability problem? The field working with UEMs has primarily studied phenomena that lack a stable and fundamental framework. We operate with usability problems without having reliable or valid definitions on what a problem is. We operate with severity of usability problems without knowing how to measure or extract severe problems. To generate more coherent, scientific results in the field of UEM, we desperately need a fundamental framework from which to work from in the future. We hope our work has enlightened the reader and perhaps given the HCI community a contribution worth reflecting on.

List of Abbreviations

The following abbreviations are used in this thesis.

ACM	Association for Computing Machinery
CW	Cognitive Walkthrough (a UEM)
GOMS	Goals, Operations, Methods, Selections (user modeling technique)
HCI	Human-Computer Interaction
HCII	Human Computer Interaction Institute (Carnegie Mellon University, Pittsburgh, US)
KLM	Key-stroke Model (user modeling technique)
MHP	Model Human Processor
PDR	Problem description report
SDA	Sequential Data Analysis
UEM	Usability Evaluation Method
UEMs	Usability Evaluation Methods
UT	Usability Test (a UEM)

Bibliography

Anderson, J.R. (1983). The architecture of cognition. Cambridge, MA: Harvard University Press.

Bærentsen, K.B., & Slavensky, H. (1999). A contribution to the design process. Communications of the ACM, 42(5), 73-77.

Bailey, G. (1993). Iterative methodology and designer training in human-computer interface design. In S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel, & T. White (Eds.), Proceedings of InterCHI'93: New York: ACM.

Bailey, R.W., Allan, R.W., & Raiello, P. (1992). Usability Testing vs. Heuristic Evaluation: A Head-To-Head Comparison. In Proceedings of the Human Factors Society 36th Annual Meeting.

Barker, R.T.,& Biers, D.W. (1994). Software usability testing: Do user self-consciousness and the laboratory environment make any difference. In Proceedings of the Human Factors and Ergonomics Society 38th Annual Meeting.

Barnard, P. (1991). Bridging between Basic Theories and the Artifacts of Human-Computer Interaction. In J. M. Carroll (Ed.), Designing Interaction: Psychology at the Human-Computer Interface. (pp. 103-127). Cambridge University Press.

Bawa, J. (1994). Comparative usability measurement: the role of the usability lab in PC Magazine UK and PC/Computing. Behaviour & Information Technology, 13(1 and 2), 17-19.

Bernadin, H.J., & Pence, E.C., (1980). Effects of rater training: new response sets and decreasing accuracy. J. Applied Psychology 65, 60-66.

Bias, R. (1994). The Pluralistic Usability Walkthrough: Coordinated Empathies. In J. Nielsen & R. Mack (Eds.) Usability Inspection Methods. 63-76, John Wiley, 1994.

Blanchard, H.E. (1998). Standards for Usability Testing. SIGCHI Bulletin, 30(3), 16-17.

Borgholm, T., & Madsen, K.H. (1999). Cooperative Usability Practices. Communications of the ACM, 42(5), 91-97.

Borman, W.C. (1979). Format and training effects on rating accuracy and rater error. J Applied psychology 64, 410-421.

Brewer, W.F., & Chinn, C.A. (1994). The theory-ladenness of data: an experimental demonstration. Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society, Hillsdale, NJ: Erlbaum, 61-65.

Brooks, P. (1994). Adding Value to Usability Testing. In J. Nielsen & R. Mack (Eds.), Usability Inspection Methods. (pp. 255-271). Wiley.

Buur, J., & Bagger, K. (1999). Replacing Usability Testing with User Dialogue. Communications of the ACM, 42(5), 63-66.

Card, S.K., Moran, T.P., & Newell, A. (1980a). Computer Text-Editing: An Information-Processing Analysis of a Routine Cognitive Skill. Cognitive Psychology, 12, 32-74.

Card, S.K., Moran, T.P., & Newell, A. (1980b). The Keystroke-Level Model for User Performance Time with Interactive Systems. Communications of the ACM, 23(7), 396-410.

Card, S.K., Moran, T., & Newell, A. (1983). The Psychology of Human-Computer Interaction. New Jersey: Lawrence Earlbaum.

Carroll, J.M., & Campbell, R.L. (1986). Softening up hard science: Reply to Newell & Card. Human-Computer Interaction. Vol. 2, 227-249.

Catani, M.B.,& Biers, D.W. (1998). Usability Evaluation and Prototype Fidelity: Users and usability Professionals. In Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting. Santa Monica, CA.

Cone, J.D. (1977). The relevance of reliability and validity for behavioral assessment. Behavioral therapy 8, 411-426.

Connell, I.W.,& Hammond, N.V. (1999). Comparing Usability Evaluation Principles with Heuristics: Problem Instances Versus Problem Types. In Proceedings of Interact'99.

Cuomo, D.L., & Bowen, C.D. (1994). Understanding usability issues addressed by three user-system interface evaluation techniques. Interacting with Computers, 6(1), 86-108.

DeHaemer, M.J. (1991). Vulcans, Klimgons and Humans: The relevance of individual differences for information systems interfaces. In ACM CPR'91

Denning, S., Hoiem, D., Simpson, M., & Sullivan, K. (1990). The Value of Thinking-Aloud Protocols in Industry: A Case Study At Microsoft Corporation. In Proceedings of the Human Factors Society 34th Annual Meeting.

Desurvire, H.W. (1994). Faster, Cheaper!! Are Usability Inspection Methods as Effective as Empirical Testing? In J. Nielsen & R. Mack (Eds.), Usability Inspection Methods. (pp. 173-201). Wiley.

Desurvire, H.W., Kondziela, J.M., & Atwood, M.E. (1992). What is Gained and Lost when using Evaluation Methods other than Empirical Testing. In A. F. Monk, D. Diaper, & M. D. Harrison (Eds.), People and Computers VII: Cambridge: Cambridge University Press.

Dillon, A., Sweeney, M., & Maguire, M. (1993). A Survey of Usability Engineering Within the European IT Industry - Current Practise and Needs. In J. L. Alty, D. Diaper, & S. Guest (Eds.), People and Computers VIII: Cambridge: Cambridge University Press.

Dillon, A., & Watson, C. (1996). User analysis in HCI - the historical lessons from individual differences research. International Journal of Human-Computer Studies, 45, 619-637.

Dix, A., Finley, J., Abowd, G., & Beale, R. (1998). Human-Computer Interaction. Second edition. Prentice Hall Europe.

Dolan, W.R., & Dumas, J.S. (1999). A Flexible Approach to Third-Party Usability. Communications of the ACM, 42(5), 83-85.

Dumas, J.S. (1989). Stimulating change through usability testing. SIGCHI Bulletin, 21(1), 37-44.

Duncker, K. (1945). On problem solving. Psychological Monographs (Whole number 270), 1-113.

Dutt, A., Johnson, H., & Johnson, P. (1994). Evaluating Evaluation Methods. In G. Cockton, S. W. Draper, & G. R. S. Weir (Eds.), People and Computers IX: 9. Cambridge: Cambridge University Press.

Egan, D.E. (1988). Individual differences in human-computer interaction. In M. Helander (Ed.) Handbook of Human-Computer Interaction. Elsevier Science Publishers B.V. (North-Holland).

Engel, S.E., & Granda, R.E. (1975). Guidelines for Man/Display Interfaces. IBM Tech. Report 00.2720. Poughkeepsie Laboratory.

Ereback, A.-L.,& Höök, K. (1994). Using Cognitive Walkthrough for Evaluating a CSCW Application. In C. Plaisant, T. Landauer, & W. Mackey (Eds.), Proceedings of CHI'94 - Conference Companion: Boston, Massachusetts: ACM.

Ericsson, K.A., & Simon, H.A. (1980). Verbal reports as data. Psychological Review 87(3), 215-251.

Gardner, J. (1999). Strengthening the focus on users' working practices; Getting beyond traditional usability testing. Communications of the ACM, 42(5), 79-82.

Gleitman, H. (1995). Psychology. W.W. Norton & Company.

Gould, J., & Lewis, C. (1985). Designing for usability: Key principles and what designers think. Communications of the ACM, (March 1985) vol. 28, no. 3, 300-311.

Gould, J.D. (1988). How to Design Usable Systems. In M. Helander (Ed.), Handbook of Human-Computer Interaction. (pp. 757-789). Elsevier Science Publishers B.V.

Gray, W.D., John, B.E., & Atwood, M.E. (1993). Project Ernestine: Validating a GOMS Analysis for Predicting and Explaining Real-World Task Performance. Human-Computer Interaction, 8, 237-309.

Gray, W.D., & Salzman, M.C. (1998). Damaged Merchandise? A Review of Experiments That Compare Usability Evaluation Methods. Human-Computer Interaction, 13(3), 203-261.

Grudin, J., & Barnard, P. (1984). The cognitive demands of learning and representing command names for text editing. Human Factors. Vol. 26, 407-422.

Hackman, G.S.,& Biers, D.W. (1992). Team usability testing: Are two heads better than one? In Proceedings of the Human Factors Society 36th Annual Meeting.

Hammond, N., Hinton, G., Barnard, P., MacLean, A., Long, J., & Whitefield, A. (1984). Evaluating the interface of a document processor: A comparison of expert judgment and user observation. In B. Shackel (Ed.), Human-Computer Interaction - INTERACT'84: North-Holland: Elsevier Science Publishers B.V.

Heckel, P. (1984). The elements of friendly software design. New York: Warner Books.

Helander, M., Landauer T. K., & Prabhu, P. (1987). (Eds.) Handbook of Human-Computer Interaction. Elsevier Science B.V.

Held, J.E.,& Biers, D.W. (1992). Software usability testing: Do evaluator intervention and task structure make any difference? In Proceedings of the Human Factors Society 36th Annual Meeting:

Henderson, R., Podd, J., Smith, M., & Varala-Alvarez, H. (1995). An examination of four user-based software evaluation methods. Interacting with Computers, 7(4), 412-432.

Hertzum, M.,& Jacobsen, N.E. (1998). The Evaluator Effect during First-Time Use of the Cognitive Walkthrough Technique. Unpublished Manuscript.

Hertzum, M.,& Jacobsen, N.E. (1999). The Evaluator Effect during First-Time Use of the Cognitive Walkthrough Technique. In H. Bullinger & J. Ziegler (Eds.), Human-Computer Interaction – Ergonomics and User Interfaces. Vol. 1., 1063-1067.

Hilbert, D., & Redmiles, D. (1998). An Approach to Large-Scale Collection of Application Usage Data Over the Internet. Proceedings of the Twentieth International Conference on Software Engineering (ICSE Ô98, Kyoto, Japan), IEEE Computer Society Press, City, April 19-25, pp. 136-145.

Holleran, P.A. (1991). A methodological note on pitfalls in usability testing. Behaviour & Information Technology, 10(5), 345-257.

Hynninen, T. Personal communication, Helsinki, September, 1999.

Jacobsen, N.E. (1996). Evaluering af kognitive gennemgang – refleksion over en konkret evaluering af CW (in Danish). Unpublished manuscript.

Jacobsen, N.E. (1997). Learning and using the cognitive walkthrough technique. Unpublished manuscript.

Jacobsen, N.E. (1999). Description of the structure diagram for the CW case study database. Technical report. Department of Psychology, University of Copenhagen, Denmark.

Jacobsen, N.E., Hertzum, M., & John, B.E. (1998b). The evaluator effect in usability studies: problem detection and severity judgments. In Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting: Santa Monica, CA: Human Factors and Ergonomics Society, 1336-1340.

Jacobsen, N.E., Hertzum, M., & John, B.E. (1998a). The evaluator effect in usability tests. In C.-M. Karat & A. Lund (Eds.), Human Factors in Computing Systems CHI'98 Summary: ACM, 255-256.

Jacobsen, N.E., & John, B.E. (1999). A tale of two critics: Case studies in using cognitive walkthrough. Paper in review. (Submitted to *Human-Computer Interaction*)

Jacobsen, N.E., & Jørgensen, A.H. (1998). Evaluating Usability Evaluation Methods. Peer-reviewed and presented at the Basic Research Symposium in conjunction with The Human Factors in Computing Systems CHI'98 Conference, Los Angeles, April, 1998.

Jacobsen, N.E., & Jørgensen, A.H. (1999). The Science of Usability Evaluation Methods in a Kuhnian Perspective. Paper in review. (Submitted to the *Human Factors and Ergonomics Society 44th Annual Meeting*)

Jeffries, R. (1994). Usability Problem Reports: Helping Evaluators Communicate Effectively with Developers. In J. Nielsen & R. Mack (Eds.), Usability Inspection Methods. (pp. 273-295). John Wiley & Sons, Inc.

Jeffries, R., Miller, J.R., Wharton, C., & Uyeda, K.M. (1991). User Interface Evaluation in the Real World: A Comparison of Four Techniques. In S. P. Robertson, G. M. Olson, & J. S. Olson (Eds.), Human Factors in Computing Systems CHI'91: New York: ACM.

John, B.E., & Kieras, D.E. (1996). Using GOMS for User Interface Design and Evaluation: Which Technique? ACM Transactions on Human-Computer Interaction, 3(4), 287-319.

John, B.E., & Mashyna, M.M. (1997). Evaluating a Multimedia Authoring Tool. Journal of the American Society for Information Science, 48(9), 1004-1022.

John, B.E.,& Packer, H. (1995). Learning and Using the Cognitive Walkthrough Method: A case study Approach. In I. Katz, R. Mack, L. Marks, M. B. Rosson, & J. Nielsen (Eds.), Proceedings of CHI'95: New York: ACM.

Jordan, P.W. (1998). An introduction to usability. Taylor & Francis Ltd.

Jordan, P.W.,& Thomas, D.B. (1994). Ecological validity in laboratory based usability evaluations. In Proceedings of the Human Factors and Ergonomics Society 38th Annual Meeting

Jørgensen, A.H. (1989). Using the Thinking-Aloud Method in System Development. In G. Salvendy & M. J. Smith (Eds.), Designing and Using Human-Computer Interfaces and Knowledge Based Systems. (pp. 743-750). Amsterdam: Elsevier Science Publishers B.V.

Jørgensen, A.H. (1990). Thinking-aloud in user interface design: a method promoting cognitive ergonomics. Ergonomics, 33(4), 501-507.

Jørgensen, A.H. & Jacobsen, N.E. (1998). Taking stock of what we know and do not know about usability evaluation methods. In Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting: Santa Monica, CA: Human Factors and Ergonomics Society , 1638.

Karat, C.-M. (1994). A Comparison of User Interface Evaluation Methods. In J. Nielsen & R. Mack (Eds.), Usability Inspection Methods. (pp. 203-233). Wiley.

Karat, C.-M., Campbell, R., & Fiegel, T. (1992). Comparison of empirical testing and walkthrough methods in user interface evaluation. In P. Bauersfeld, J. Bennett, & G. Lynch (Eds.), Proceedings of CHI'92: New York: ACM.

Keister, R.S. (1981). Human Factors guidelines for the content, layout, and format of CRT displays in interactive application systems. Human Factors Guidelines Series, NCR internal report. HFP 81-08 (Version No. 3).

Kieras, D.E., & Polson, P.G. (1985). An approach to the formal analysis of user complexity. Int. J. of Man-Machine Studies, 22, 365-394.

Kiger, J.F., (1984). The depth/breath trade-off in the design of menu-driven user interfaces. Int. J. Man-Machine Studies. Vol. 20, 201-213.

Kirakowski, J. Personal communication, October, 1998.

Kirakowski, J. & Dillon, A. (1988). The computer user satisfaction inventory (CUSI): Manual and scoring key. Cork, Ireland: Human Factors Research Group, University College of Cork.

Knowles, C. (1988). Can cognitive complexity theory (CCT) produce an adequate measure of system usability? In People and Computers IV. University Press.

Kurosu, M., Sugizaki, M., & Matssura, S. (1999). A Comparative Study of sHEM (structed heuristic evaluation method). In H. Bullinger & J. Ziegler (Eds.), Human-Computer Interaction – Ergonomics and User Interfaces. Vol. 1., 938-942.

Laird, J.E., Newell, A., & Rosenbloom, P.S. (1987). Soar: An Architecture for General Intelligence. Artificial Intelligence, 33, 1-64.

Landauer, T.K. (1987). Relations between Cognitive Psychology and Computer System Design. In J. M. Carroll (Ed.), Interfacing Thought: Cognitive Aspects of Human-Computer Interface. (pp. 1-25). London: Bradford/MIT Press.

Landauer, T.K. (1997). Behavioral Research Methods in Human-Computer Interaction. In M. Helander, T. K. Landauer, & P. Prabhu (Eds.), Handbook of Human-Computer Interaction. (pp. 203-227). Elsevier Science B.V.

Lavery, D., Cockton, G., & Atkinson, M. (1997). Comparison of evaluation methods using structured usability problem reports. Behaviour & Information Technology, 16(4/5), 246-266.

Lewis, C. (1982). Using the "Thinking-aloud" Method in Cognitive Interface Design. Yorktown Heights, NY: IBM Thomas J. Watson Research Center. RC 9265 (#40713).

Lewis, C. (1988). How and why to learn why: analysis-based generalization of procedures. Cognitive Science, 12, 211-256.

Lewis, C., Polson, P.G., Wharton, C., & Rieman, J. (1990). Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces. In J. C. Chew & J. Whiteside (Eds.), Proceedings of CHI'90: New York: ACM.

Lewis, C., & Wharton, C. (1997). Cognitive Walkthrough. In M. Helander, T. K. Landauer, & P. Prabhu (Eds.), Handbook of Human-Computer Interaction. (pp. 717-732). Elsevier Science B.V.

Lewis, J.R. (1991). Psychometric evaluation of an after-scenario questionnaire for computer usability studies: the ASQ. SIGCHI Bulletin, 23(1), 78-81.

Lewis, J.R. (1994). Sample Sizes for Usability Studies: Additional Considerations. Human Factors, 36(2), 368-378.

Licklider, J.C.R. (1960). Man-Computer Symbiosis. IRE Transactions on Human Factors in Electronics, (March), 4-11.

Lord, R.G. (1985). Accuracy in behavioral assessment: an alternative definition based on raters' cognitive schema and signal detection theory. J Applied psychology 70(1), 66-71.

Mack, R., & Montaniz, F. (1994). Observing, Predicting, and Analyzing Usability Problems. In J. Nielsen & R. Mack (Eds.), Usability Inspection Methods. (pp. 295-339). Wiley.

Mack, R., & Nielsen, J. (1993). Usability Inspection Methods: Report on a Workshop Held at CHI'92, Monterey, CA, May 3-4, 1992. SIGCHI Bulletin, 25(1), 28-33.

May, J., & Barnard, P. (1995). The case for supportive evaluation during design. Interacting with Computers, 7(2), 115-143.

Mills, C.B. (1987). Usability testing in the real world. SIGCHI Bulletin, 19(1), 43-46.

Molich, R. (1994). Preventing user interface disasters. Behaviour & Information Technology, 13(1 and 2), 154-159.

Molich, R., Bevan, N., Curson, I., Butler, S., Kindlund, E., Miller, D., & Kirakowski, J. (1998). Comparative Evaluation of Usability Tests. In Proceedings of UPA98 (Usability Professionals Association 1998 Conference).

Molich, R., & Nielsen, J. (1990). Improving a Human-Computer Dialogue. Communications of the ACM, 33(3), 338-348.

Muller, M.J., & Czerwinski, M. (1999). Organizing Usability Work To Fit the Full Product Range. Communications of the ACM, 42(5), 87-90.

Newell, A., & Card, S.K. (1985). The Prospects for Psychological Science in Human-Computer Interaction. Human-Computer Interaction, 1, 209-242.

Newell, A., & Simon, H.A., (1972). Human Problem Solving. Englewood Cliffs, NJ: Prentice-Hall.

Newman, W., & Lamming, M.G. (1995). Interactive System Design. Addison-Wesley Publishing Company.

Nielsen, J. (1992). Finding usability problems through heuristic evaluation. In P. Bauersfeld, J. Bennett, & G. Lynch (Eds.), Proceedings of CHI'92: New York: Addison-Wesley.

Nielsen, J. (1993). Usability Engineering. Boston: Academic Press Inc.

Nielsen, J. (1994). Estimating the number of subjects needed for a thinking aloud test. International Journal of Human-Computer Studies, 41, 385-397.

Nielsen, J. (1995). Getting usability used. In K. Nordby, P. Helmersen, D. J. Gilmore, & S. Arnsen (Eds.), Human-Computer Interaction: Interact'95: Chapman and Hall.

Nielsen, J., & Landauer, T.K. (1993). A Mathematical Model of the Finding of Usability Problems. In S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel, & T. White (Eds.), Proceedings of InterCHI'93: New York: ACM.

Nielsen, J., & Mack, R.L., (1994). Usability Inspection Methods. John Wiley & Sons, Inc.

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. In J. C. Chew & J. Whiteside (Eds.), Proceedings of CHI'90: New York: Addison-Wesley.

Nielsen, J.,& Phillips, V. (1993). Estimating the relative Usability of Two Interfaces: Heuristic, Formal, and Empirical Methods Compared. In S. Ashlund, A. Henderson, E. Hollnagel, & T. White (Eds.), Proceedings of InterCHI'93: New York: ACM.

Nisbett, R.E., & Wilson, T.D. (1977). Telling more than we can know: verbal reports on mental processes. Psychological review, 84(3), 231-259.

Norman, D.A. (1988). The Design of Everyday Things. New York: Doubleday.

Norman, D.A. (1995). On Differences Between Research and Practice. Ergonomics in Design, 35-36.

Norman, D.A., & Draper, S.W. (1986). User Centered Systems Design. Hillsdale, N.J.: Erlbaum.

Ohnemus, K.R.,& Biers, D.W. (1993). Retrospective versus concurrent thinking-out-loud in usability testing. In Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting: Vol. 2.

Olson, J.S., & Moran, T.P. (1996). Mapping the Method Muddle: Guidance in Using Metods for User Interface Design. In M. Rudisill, C. Lewis, P. G. Polson, & T. D. McKay (Eds.), Human-Computer interface design: Success stories, emerging methods, and real world context. (pp. 269-299). San Francisco: Morgan Kaufmann Publishers.

Olson, G.M., & Moran, T.P. (1998). Introduction to This Special Issue on Experimental Comparisons of Usability Evaluation Methods. Human-Computer Interaction, 13(3), 199-201.

Pane, J.F., & Miller, P.L. (1993). The ACSE multimedia science learning environment. In Proceedings of the 1993 International Conference in Computers in Education, Taipei, Taiwan.

Pejtersen, A. M., & Rasmussen, J. (1997). Effectiveness testing of complex systems. In G. Salvendy (Ed.) Handbook of Human Factors and Ergonomics (pp. 1514-1542). New York: John Wiley

Perstrup, K., Frøkjær, E., Konstantinovitz, T., Sørensen, F.S., Varmning, J. (1997). A World Wide Web-based HCI-library designed for interaction studies. In Third ERCIM user interfaces for all workshop (Obernai, France, November, 1997).

Polson, P.G., & Lewis, C. (1990). Theory-Based Design for Easily Learned Interfaces. Human-Computer Interaction, 5, 191-220.

Polson, P.G., Lewis, C., Rieman, J., & Wharton, C. (1992). Cognitive walkthrough: a method for theory-based evaluation of user interfaces. International Journal of Man-Machine Studies, 36, 741-773.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey, T. (1994). Human-Computer Interaction. Addison-Wesley.

Rasmussen, J. (1983). Skills, rules, knowledge: signals, signs and symbols and other distinctions in human performance models. IEEE Transactions: Systems, Man & Cybernetics, SMC-13, 257-267.

Reason, J. (1990). Human Error. Cambridge University Press.

Rosenthal, R. (1963). On the social psychology of psychological experiments: the experimenters hypothesis as unintended determinant of experimental results. American Scientist, 51, 261-283.

Rowley, D.E.,& Rhoades, D.G. (1992). The Cognitive Jogthrough: A Fast-Paced User Interface Evaluation Procedure. In P. Bauersfeld, J. Bennett, & G. Lynch (Eds.), Proceedings of CHI'92: New York: ACM.

Rubin, J. (1994). Handbook of usability testing: how to plan, design, and conduct effective tests. John Wiley & Sons, Inc.

Salzman, M.C., & Rivers, S.D. (1994). Smoke and mirrors: setting the stage for a successful usability test. Behaviour & Information Technology, 13(1 and 2), 9-16.

Sanderson, P., Scott, J., Johnston, T., Mainzer, J., Watanabe, L., & James, J. (1994). MacSHAPA and the enterprise of exploratory sequential data analysis (ESDA). International Journal of Human-Computer Studies, (41), 633-681.

Scarr, S. (1985). Constructing psychology: making facts and fables for our times. American Psychologist 40(5), 499-512.

Schmid, R.E. (1998). http://www.washingtonpost.com/wp-srv/national/daily/july/18/twa.htm

Shackel, B. (1991). Usability - context, framework, definition, design and evaluation. In B. Shackel., & S. J. Richardson (Eds) Human Factors for Informatics Usability. 1991. Cambridge University Press, 21-37.

Shneiderman, B., (1998). Designing the user interface: strategies for effective human-computer interaction. Third edition. Addison-Wesley.

Smilowitz, E.D., Darnell, M.J., & Benson, A.E. (1993). Are We Overlooking Some Usability Testing Methods? A Comparison of Lab, Beta, and Forum Tests. In Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting.

Smith, S.L., & Mosier, J.N. (1986). Guidelines for designing user interface software. Mitre Corporation Report MTR-9420, Mitre Corporation, 1986.

Sullivan, K. (1996) The Windows95 User Interface: A Case Study in Usability Engineering. In Human Factors in Computing Systems CHI'96 Conference Proceedings: ACM.

Thorndike, R.L. (1954). The psychological value systems of psychologists. American Psychologist, 9, 787-789.

Tversky, A., & Kahneman, D. (1974). Judgment under Uncertainty: Heuristics and Biases. Science, 185, 1124-1131.

Vainio-Larsson, A.,& Orring, R. (1990). Evaluating the usability of user interfaces: research in practice. In D. Diaper (Ed.), Proceedings of IFIP INTERACT'90: Human-Computer Interaction 1990: Elsevier Science Publishers B.V. (North-Holland).

Virzi, R.A. (1990). Streamlining the Design Process: Running Fewer Subjects. In Proceedings of the Human Factors Society 34th Annual Meeting.

Virzi, R.A. (1992). Refining the Test Phase of Usability Evaluation: How Many Subjects Is Enough. Human Factors, 34(4), 457-468.

Virzi, R.A., Sorce, J.F., & Herbert, L.B. (1993). A Comparison of Three Usability Evaluation Methods: Heuristic, Think-Aloud, and Performance Testing. In Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting.

Wenger, M.J., & Spyridakis, J.H. (1989). The Relevance of Reliability and Validity to Usability Testing. Transactions on Professional Communication, 32(4), 265-271.

Wharton, C., Bradford, J., Jeffries, R., & Franzke, M. (1992). Applying cognitive walkthroughs to more complex user interfaces: Experiences, issues, and recommendations. In P. Bauersfeld, J. Bennett, & G. Lynch (Eds.), Procedings of CHI'92: New York: ACM.

Wharton, C., Rieman, J., Lewis, C., & Polson, P.G. The Cognitive Walkthrough Method: A Practitioner's Guide. (1993). Institute of Cognitive Science, University of Colorado, Boulder, Colorado: #CU-ICS-93-07.

Wharton, C., Rieman, J., Lewis, C., & Polson, P.G. (1994). The Cognitive Walkthrough Method: A Practitioner's Guide. In J. Nielsen & R. Mack (Eds.), Usability Inspection Methods. (pp. 105-140). Wiley.

Wright, P.C., & Monk, A.F. (1991a). The use of think-aloud evaluation methods in design. SIGCHI Bulletin, 23(1), 55-57.

Wright, P.C., & Monk, A.F. (1991b). A cost-effective evaluation method for use by designers. International Journal of Man-Machine Studies, 35, 891-912.

Yin, R.K. (1994). Case Study Research - Design and Methods. (Second Edition ed.). Thousand Oaks, California: SAGE Publications.

Zhang, Z., Basili, V., & Sneiderman, B. (1998). An empirical study of perspective-based usability inspection. In Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting: Santa Monica, CA: Human Factors and Ergonomics Society, 1346-1350.

Zirkler, D., & Ballman, D. (1994). Usability testing in a competitive market: lessons learned. Behaviour & Information Technology, 13(1 and 2), 191-197.

Zobbe, A. Personal communication. August, 1999.

Zunde, P., & Dexter, M. (1969). Indexing consistency and quality. American Documentation 20(3) July.

Appendices