

Prediction of Usability: Comparing Method Combinations

Erik Frøkjær

University of Copenhagen
Department of Computing
Universitetsparken 1
DK-2100 Copenhagen, Denmark
+45 35 32 14 56
fax: +45 35 32 14 01
erikf@diku.dk

Marta K. Lárusdóttir

EJS hf.
Grensásvegi 10
108 Reykjavík
Iceland
+354 - 563 30 00
fax: +354- 563 84 13
marta@ejs.is

ABSTRACT

The effectiveness of three methods for uncovering and assessing usability problems has been derived from usage reports from 17 groups of evaluators with 3 evaluators per group. The evaluators were third year computer science students. The methods investigated were Cognitive Walkthrough, Heuristic Evaluation, and Thinking Aloud. The effect of combined use of two evaluation methods were investigated by first doing either Cognitive Walkthrough or Heuristic Evaluation followed by Thinking Aloud. When used alone, Heuristic Evaluation detects significantly more problems than Cognitive Walkthrough. Thinking Aloud after prior use of Heuristic Evaluation seems superior in detecting usability problems when compared to any other methods, and it also eliminates an important weakness of Heuristic Evaluation when used by non-experts, namely its proneness to indicating false problems.

Keywords

Usability evaluation, experimentation, method combination, predictive power, effectiveness, Heuristic Evaluation, Cognitive Walkthrough, Thinking Aloud.

1 INTRODUCTION

The purpose of this study is to compare the effectiveness of predicting usability problems by three widely taught and applied usability evaluation methods (UEMs). Usability problems are considered to be those that will influence the users of the system during practical use. The three UEMs selected for the study are Thinking Aloud (TA) [15, 18] and the two inspection methods, Heuristic Evaluation (HE) [20, 18] and Cognitive Walkthrough (CW) [22, 25]. For the first time, the effect of combined use of two evaluation methods are investigated by first doing one of the inspection methods, then doing TA, on the same user interface. Further, we have examined whether the UEMs tend to waste efforts by addressing so-called *false problems*—a false problem being a usability problem registered that in practice influences no or very few users. For each method, the time spent by the evaluators learning the techniques, and preparing and doing the evaluation has been recorded. The evaluators were third year computer science students in the final semester of their bachelor program. The number of evaluators in this study was 51; higher than in any other previously published study of the effectiveness of UEMs.

Since the early nineties researchers have carried out studies comparing and contrasting some of the methods brought forward to uncover usability problems of interactive computer systems, e.g. [8, 3, 14, 2, 13]. These studies still have to be considered preliminary as pointed out by John and Mashyna [11], Gray and Salzman [5], and Baecker et al. [1: p. 87], the latter asking for studies where combinations of inspection methods and empirical usability testing are evaluated. The

The term a *usability experiment* is here used as a common designation of either an empirical usability evaluation, here TA, or an inspection based usability evaluation, here HE or CW.

2 THE DESIGN OF THE EXPERIMENT

Evaluation Methods Selected

The three UEMs, HE, CW and TA, selected for this method evaluation were introduced uniformly to the evaluators. They received descriptions of HE by [18: pp. 19-20 and pp. 115-165]. CW was introduced by [18: Chapter 5]. TA was introduced by [18: pp. 181-191 and pp. 195-198] and by [15: pp. 51-56].

CW and HE were selected because these two inspection methods seem to be the most widely taught and used in Europe and in the United States. TA is a very common method for usability testing, and we knew from previous studies [12, 16] that computer science students and system designers are able to make quite effective use of this method without heavy training.

We decided to make use of more recently published descriptions of the methods than were available to earlier prominent method evaluation studies, e.g. [8, 14, 3, 2], although we emphasized to use introductory descriptions prepared by the original authors of the methods. Direct comparisons with results from earlier studies were hereby made problematic, but it would so for a number of other reasons, see below in the section Comparative Discussion.

The Method Evaluators

The evaluators were 51 students in their third year of a bachelor degree in computer science. The evaluators' mean age was 25.1 year; 85% were male. 30% of the evaluators was or had been working in industry as software designers or programmers, typically on a part time basis. Participation in this method evaluation project was a mandatory, grade-giving part of a semester course in Human-Computer Interaction. The students could decide whether the researchers were allowed to include their results in this comparative study. All collected data were anonymously related to the individual evaluator and his or her group. Nearly half of the evaluators had heard about TA before, but less than five percent had ever used it. Only 10% had heard about CW and HE, none had used any of them.

The general computing and system development skills of the student evaluators are not too far from the typical level of programmers and software developers in industry in Scandinavia, and probably in many parts of Europe. This statement is based on the fact that these third year students already have, or easily obtain, responsible positions as software developers.

Software Tools Selected for Evaluation

The evaluators evaluated either one or two of the following two software tools for usability problems: (1) An experimental text retrieval system TeSS giving access to documentation of certain programming tools [6], or (2) the graphical text editor Asedit, a Unix-based shareware program developed by Andrzej Stochniol. The evaluators received a short user manual about TeSS. Asedit includes hypertext on-line help facilities and user documentation. TeSS and Asedit are quite simple systems to an audience like the evaluators, and could be used for practical tasks after a few half hours' introductory training. Only 10% of the evaluators had heard about TeSS, none had used it. None had heard about Asedit.

The Experimental Procedure

During a two-week period 17 groups of 3 evaluators performed 28 usability experiments following one out of three evaluation programs, see . 1. The evaluator groups were randomly assigned to one of the evaluation programs each starting by evaluating one of the two software tools using one of the two inspection methods. The second week all groups evaluated the same software tool TeSS using the TA method.

Table 1. The procedure of the experimental method evaluation.

<i>Preparation</i>	Voluntary group formation, 51 students made 17 groups, 3 individuals in each group. The groups were randomly assigned to one of three evaluation programs.		
<i>Evaluation Program no.</i>	1	2	3
<i>Evaluation name</i>	CW 6 groups evaluated TeSS using CW	HE 5 groups evaluated TeSS using HE	Control 6 groups evaluated Asedit, 3 using CW and 3 using HE
<i>1. Week</i>			
<i>Result</i>	6 LUPs	5 LUPs	6 LUPs (not used)
<i>Evaluation name</i>	CW-TA 6 groups evaluated TeSS using TA	HE-TA 5 groups evaluated TeSS using TA	TA 6 groups evaluated TeSS using TA
<i>2. Week</i>			
<i>Result</i>	6 LUPs	5 LUPs	6 LUPs
<i>Data Analysis</i>	The authors combined the 28 LUPs on TeSS (6 from CW, 5 from HE and 17 from TA) for making the A-LUP list for TeSS		

After one week's work with one of the inspection methods each group submitted a List of Usability Problems (LUP) to the research group, (see description below). After still one week the groups submitted the LUPs based upon TA.

By leaving out Asedit from the TA evaluation we obtained to have 6 groups to meet TeSS for the first time (Evaluation Program 3 in Table 1). This could make it possible to identify, and maybe to compensate, for a learning effect—or an effect where evaluators who had first used an inspection method might be biased in their empirical testing, wanting to confirm the usability problems they had already found.

A strict procedure of the evaluations forced the students to carry out and report individual evaluations using HE before they opened discussions with their group members for reaching a common result and preparing the group's LUP, as recommended by Nielsen [18].

For the TA tests another 51 computer science students acted as subjects. Groups of 3 subjects were randomly assigned to each group of TeSS evaluators. These students had followed similar evaluation programs as the TeSS evaluators, but instead of TeSS they evaluated a Unix-based email-system, Elm, which was familiar to them. The purpose of this arrangement was two-fold: (a) to be able to assign subjects without prior knowledge of TeSS to the 17 groups for their TA sessions; and (b) to let the students experience that systematic use of UEMs would reveal a rather large number of severe usability problems, even in tools they were accustomed to. As a result of the TA tests the groups handed in a LUP for each of the 3 subjects, and a combined LUP for the TA experiment, which are used in this study.

The reasons for a group-based evaluation program were that we wanted the students to benefit from mutual discussion during their process of training the usage of either CW or HE. Also we wanted them to be able to have different roles in the TA tests, as a moderator, an observer, and a registrant. We expect a group-based evaluation program to be quite realistic for system developers. In a corporate environment Karat et al. [14] found group-based walkthroughs superior compared to individual walkthroughs.

The Data Collected

The evaluators described the results of each usability experiment in a List of Usability Problems (LUP). Each entry in such a list consisted of a problem description, and the evaluators' severity rating of the problem according to the scale in Table 2.

Table 2. The three point scale used by the evaluators to grade the severity of the usability problems.

- *Highly Critical Problems* (HCP), i.e. problems that decisively influence whether or not the user is able to perform the ongoing operation and complete the task. It is strongly recommended that such problems should be corrected before the system is put into operation.
- *Severe Problems* (SP), i.e. problems that impede the user's work substantially and have some influence on whether the user can perform the ongoing task. Such problems should at least be corrected in the next version of the system.
- *Cosmetic Problems* (CP), i.e. problems that impose only slight inconvenience to the user. Such problems should be corrected when a convenient opportunity turns up.

In this scale, the descriptions about when the problems should be corrected was only meant as an indicator of the severity of the problem. The evaluators should not in their classification decision take into consideration any other aspects than usability, i.e. not cost to fix. Severity classifications similar to the one presented in Table 2 are used in many other UEM studies.

Additionally we have collected background information about each evaluator's study activities, professional work experience, prior knowledge about the UEMs and software tools similar to the tools selected for this study.

3 DATA ANALYSIS

As a common reference for the assessment of the evaluator groups' LUPs, we formed an Authorized List of Usability Problems (A-LUP) of TeSS. This list contains all usability problems identified in the 51 evaluators' 28 usability experiments, see table 1 The A-LUP identifies each problem, describes the problem shortly, and grades the severity of the problem.

Identification of Problems

The identification of the problems of the A-LUP was done by collecting all the problem descriptions from the 28 LUPs. The problem descriptions were grouped according to the different parts of the interface of TeSS. Descriptions of identical problems were registered as one; and descriptions which covered more than one problem were split and registered accordingly. We preferred the problem descriptions in their concrete forms avoiding slipping into more general expressions of problem types. We expected concrete problem descriptions to be more useful during the appraisals of the groups' LUPs against the A-LUP.

Neutralized Problems

Some of the problems registered were not directly attributable to TeSS. Such problems were identified independently of the severity of the problem and they have been neutralized, i.e. they have not been taken into this comparative analysis. The problems neutralized were categorized as either (1) evaluator's error, (2) user's extreme and intricate manner of using TeSS, e.g. what programmers sometimes call "crash-testing", (3) not a usability problem related to TeSS, (4) not reproducible, (5) meaningless problem description, (6) problem of an underlying system, in the case of TeSS the X Window System or Unix. Table 3 contains an overview of the number of problems of each category of problems neutralized.

False Problems

Some problems from the LUPs we considered to be false problems in the sense that—if such a problem would cause change of the evaluated software tool—we would expect this change to have no or even a negative effect on the usability of the system. We registered 14 false problems. As an illustration we can mention ‘TeSS at DIKU ought to be in Danish’. We do not consider this problem to be of relevance in this particular context. The body of texts made searchable in TeSS are programming manuals written in English, and the intended users of TeSS are computer science students who are quite familiar with the English language. Further, TeSS is only available through the Unix-based computer network of the department, where practically all the other software tools offer an English-oriented user interface only. Identifying a problem as false can raise many complicated questions and discussions. We have been restrictive in using this category. For instance, there are only three false problems that have been identified by more than one LUP. Table 3 gives an overview of the usability problems registered, arranged according to the categories of the A-LUP.

Table 3. Categories of problems included by the Authorized List of Usability Problems (A-LUP) with total number of problems by each category.

107	<i>Usability Problems</i>
3	Highly Critical Problems, HCP
38	Severe Problems, SP
66	Cosmetic Problems, CP
14	<i>False Problems</i>
34	<i>Neutralized Problems</i>
8	Evaluators' error
5	User's choice of an extreme manner of operation
7	Not a usability problem related to the software tool evaluated
2	Programming error
2	Not reproducible
5	Meaningless
5	Underlying system
155	<i>Total number of identified problems</i>

The Severity of the Usability Problems of the A-LUP

A crucial point of the present study is how we reached the grading of the severity of the usability problems of the A-LUP. The accumulated results from the 17 TA experiments were taken carefully into account, but because of large variances, see section 4, the grading was not just a matter of computing the evaluators' mean grading. We decided that no problem identified by more than one TA experiment as being severe or highly critical was graded less than a SP, i.e. a noticeable usability problem. In this way we adjusted towards the evaluators' final assessments after their TA sessions avoiding disputable grading of potential SPs as CPs. In the high end, the definition of a Highly Critical Problem was so strict, that we knew quite surely what actually the HCPs of TeSS were by virtue of an earlier study with feed-back from 83 subjects working 648 hours on-line with TeSS during their solution of 25 information retrieval tasks [6].

This section first presents the results of the statistical analysis of the methods suitability to identifying the usability problems of TeSS. Then the methods suitability to revealing the proper severity of the identified problems are examined.

As background information concerning the time spent, an average evaluator, as a member of the group of evaluators, used respectively 22 hours, 29 hours and 24 hours doing a complete CW, HE or a TA. The differences are statistically significant, but not large enough to be really important. These figures include between 6-9 hours for an evaluator reading the description of the method, reaching an understanding of its techniques, and together with the other members of the evaluator group preparing the usability experiment.

The Effectiveness of the Methods to Predict Usability Problems

Our study shows a significant difference between CW and HE, and between CW and TA ($p < 0.05$ by t-tests), but the difference between HE and TA is not significant, see Table 4. With HE an average group of evaluators finds 19% of all the usability problems of TeSS, while an average group using CW finds 10%.

HE and TA performed by a group of three evaluators on an average uncovered approximately 80% more usability problems than a group using CW. The strict GOMS-like walkthrough built into CW seems to be unrealistically narrow focused compared to the users' actual activity during system interaction.

Table 4. The mean percentage of usability problems from the A-LUP found by a group in one usability experiment using the specific UEM.

Usability Evaluation Method (UEM)	CW	HE	TA
Mean percentage of the usability problems	10%	19%	18%
Standard deviation ()	4%	5%	9%

Nielsen [17] has reported that 3 developers using HE found approximately 40% of the usability problems of a software tool. He asked 31 computer science students, in their first year, to evaluate an interface in which 16 usability problems had already been identified. Our groups of 3 evaluators found only half as many problems as could be expected from Nielsen's experiment. A possible explanation could be that our system, although being rather simple, was definitely more complex than the walk-up and use interface investigated by Nielsen. Our A-LUP includes 107 usability problems, compared to Nielsen's 16. Our result indicates that more HE evaluations than usually recommended have to be carried out in order to cover the problems of a specific user interface of even small systems properly. But readers of this kind of studies have to be cautious drawing quantitative, general conclusions from the results. The specific circumstances differ markedly between the studies, so what is important is probably only the relative effectiveness of the methods within each study.

The effect of combining results of using the inspection methods and the TA method is presented in Table 5.

Table 5. The mean percentage of usability problems from the A-LUP found by a group in one TA experiment, containing three TA sessions, after prior use of CW (CW-TA) or HE (HE-TA). TA indicates the six control groups, see Table 1.

Mean percentage of the usability problems	15%	25%	18%
Standard deviation ()	7%	11%	9%

Although the figures show a quite substantial mean improvement for HE-TA as compared to CW-TA and to TA, HE-TA is not statistically significant better ($p=0.22$ by an F-test). There are important variances among the groups within each of the three evaluation programs. This indicates that the individual differences among evaluators have important effects [7], even in our group-based evaluations.

Predicting the Problem Severity

Incorrect grading of problem severity may have unpleasant effects because the resources for correcting usability problems are always limited, i.e. not all problems can be corrected; and each correction has a risk of introducing new usability problems. If evaluators assign too high a problem severity, valuable resources are wasted on correcting problems of little or no significance. If evaluators assign too low a problem severity, there is a risk that a severe problem is overlooked. A cost-conscious project manager would normally not use time to correct problems that are classified as cosmetic. Table 6 presents how the usability problems graded as HCP by the present authors were graded by an average group of evaluators. The most remarkable result is how the HCP, when found by CWs, are graded as only CP in 39% of the instances. It seems that CW leaves the evaluators with a weak feeling of problem severity.

Table 6. The Highly Critical Problems of the A-LUP as graded by the groups of evaluators.

	CW	HE	TA	CW-TA	HE-TA
<i>Graded as</i>					
HCP	17%	33%	33%	6%	53%
SP	28%	47%	17%	17%	13%
CP	39%	7%	11%	33%	13%
Total of Found HCP	83%	87%	61%	56%	80%
Not Found HCP	17%	13%	39%	44%	20%

Table 7 presents how the usability problems graded as SP by the present authors were graded by the groups of evaluators. A SP graded as HCP isn't a big problem; but if SP are graded as CP then the users will probably continue to be substantially impeded in their work with the system, because the large number of usability problems typically detected by these evaluation methods makes it necessary to give CP a very low priority during repairing or re-designing of the system. When using TA the evaluators tend to misjudge the SP as CP.

Table 7. The Severe Problems of the A-LUP as graded by the groups of evaluators.

	CW	HE	TA	CW-TA	HE-TA
<i>Graded as</i>					
HCP	3%	6%	8%	7%	9%
SP	6%	15%	10%	11%	15%
CP	4%	4%	7%	7%	12%
Total of Found SP	13%	25%	25%	25%	36%
Not Found SP	87%	75%	75%	75%	64%

Table 8 presents how the usability problems graded as CP by the present authors were graded of the groups of evaluators. Here we find a proneness to overestimate the severity of the CP by evaluators using HE. CW identifies only few CP.

Table 8. The Cosmetic Problems of the A-LUP as graded by the groups of evaluators.

	CW	HE	TA	CW-TA	HE-TA
<i>Graded as</i>					
HCP	0%	2%	1%	1%	1%
SP	1%	4%	3%	3%	5%
CP	3%	5%	8%	4%	10%
Total of Found CP	5%	12%	12%	8%	16%
Not Found CP	95%	88%	88%	92%	84%

In summary, this examination of the groups' ability to indicate properly the severity of the identified usability problems shows a mixed pattern. CW seems to lack guidance for the evaluators to grade especially the HCP realistically. HE seems to lack guidance in grading the CP properly, an important weakness because of the typical large number of CP, see Table 3.

False Problems

Table 9 presents that the three methods differ significantly with regard to identifying false problems, i.e. problems that in practice influence no or very few users. A HE experiment on the average identifies 23% of the false problems on the A-LUP, TA identifies 1%, and CW none. Most of the false problems are graded as CP, but again HE stands out with a considerable part of false problems graded as SP. This is a serious weakness of HE which can be redressed by using TA afterwards.

Table 9. False problems

	CW	HE	TA	CW-TA	HE-TA
<i>Grades as</i>					
HCP	0%	0%	1%	0%	0%
SP	0%	11%	0%	0%	0%
CP	0%	11%	0%	1%	1%
Total of Found False Problems	0%	23%	1%	1%	1%
Not Found False Problems	100%	77%	99%	99%	99%

Comparative Discussions

No previous study has investigated the effect of using two evaluation methods on the same user interface and comparing this with using only one method. The result that the combination of first doing HE then doing TA seems superior in predictive power has relevance to practitioners interested in how to maximize the effectiveness of their usability evaluation program.

Many other studies have compared the predictive power of selected UEMs. The most influential of these, according to Gray and Salzman [5], are the studies [8, 14, 17, 3, 21]. Gray and Salzman [5] found methodological flaws in all these studies and call much of what we thought we knew regarding the efficiency of various UEMs into question.

Discussion Related to Jeffries et al. (1991)

Jeffries et al. [8] studied four methods: HE, guidelines, CW (in an earlier and more complex version that we have used) and usability testing. The specific form of the usability testing used in

presumably it had similarities to our TA tests. Four user interface specialists did the HE, a team of 3 software engineers did the guideline evaluation, a team of 3 software engineers did the CW, and 6 regular PC users participated in the user testing group.

Concerning the study by Jeffries et al, Gray and Salzman [5] summarizes that (a) the conclusions regarding one UEM versus another, and (b) the claims made about the types of problems found by each UEM, are problematic. There are uncontrolled differences among the evaluator groups, and the small number of evaluators and usability experiments result in statistical insignificance.

Jeffries et al. found that HE uncovered the most problems, including more of the serious ones, and that usability test also did a good job compared to guidelines and CW. The usability testing failed to find many of the serious problems, a result that corresponds to what we found for the TA tests, see Table 6. One explanation could be that usability testing is highly dependent on the tasks used as basis for the TA process. In the method descriptions very little is explained to the evaluators about how to construct these tasks. Unless the set of test tasks is comprehensive and adequate to cover all facets of the system, it is problematic to let a number of usability tests, or TA tests, have the final word in determining the severity of the problems of a specific user interface.

Another observation in the study by Jeffries et al. is that especially HE runs the risk of finding too many problems, “some of which may not be the most important ones to correct”. This corresponds to our study, see Table 8 and 9. This is particularly a problem when the method, as in our study, is used by people who are not user interface specialists. Many ordinary system developers who will be trained and involved in usability evaluations in industry will tend to pursue many cosmetic or even false problems. Training material and probably even the procedures of HE should be modified to address this aspect more directly.

Note that CW in our study practically avoids false problems, so this method has some kind of filter towards this. Why not try to utilize this in a modified version of HE? The idea to include task scenarios during the HE is one possibility used by Karat et al. [14]—another idea strongly suggested by the results of the present study is to combine TA testing into HE in some way.

We do not find an improvement when CW is followed by TA as compared to just using TA. This might not be too surprising. The approaches of CW and TA have clear similarities by virtue of the predefined tasks to be studied. You could look upon CW as a simulated TA evaluation focused by a GOMS-like walkthrough. But HE and CW represent different filters towards the user interaction, as witnessed empirically in our study and in other studies, e.g. [2]. This leads to the idea of intertwining some kind of heuristic walkthrough with CW. It would be interesting to see such methods developed and empirically tested.

Discussion Related to Desurvire et al. (1992)

In Desurvire et al. [3] groups of evaluators did usability evaluations using CW, HE and a laboratory usability test. Their study focused on how the selected methods functioned as tools for evaluators with different types of expertise, i.e. human factors experts, users of computer systems and the software engineers of the system under evaluation. The HE and the CW groups used “paper flow-charts organized by task” to complete six tasks. The user testing group used a prototype of the interface, according to information from H. Desurvire to Gray and Salzman [5]. The user testing group had 18 participants; the six inspection UEM groups had 3 participants each, but the 3 software engineers participated in both HE and CW.

Concerning the study by Desurvire et al., Gray and Salzman [5] summarizes: “The prerequisites for an experimental study—statistical conclusion validity and internal

validity—were severely lacking. ... We believe that there is nothing that can be safely concluded regarding UEMs or expertise based on this study.”

The design of the experimental method evaluation used by Desurvire et al. differs markedly from ours making it meaningless to go into detailed comparisons. The main conclusion in their study, that a laboratory based usability test is clearly the most effective to uncover the problems, cannot be compared to our results. Our TA tests are much more restricted in the approach and the invested efforts than the laboratory usability testing of the Desurvire et al. study. They use human factor experts to do the testing and collect the results from 18 users to establish their ‘authorized’ usability list. We use students with knowledge and experience comparable to well-informed, non-expert system developers in industry to make or supervise the usability evaluations. On the other hand, we draw results from 51 evaluators. The indication of the Desurvire et al. study [3]—that HE is better than CW to predict usability problems, especially when used by human factor experts—was confirmed in our study with computer science students as evaluators.

Discussion Related to Karat et al. (1992)

The Karat et al. [14] compared user testing with a walkthrough technique that combined scenarios with guidelines, i.e. a set of 12 usability heuristics. There were 48 participants in the study. They were predominantly end users and developers of GUI systems, along with a few UI specialists and software support staff. The participants were randomly assigned to three method conditions: user testing (two groups of 6 individuals), individual walkthrough (two groups of 6 individuals) and team walkthrough (two groups with six teams of 2 individuals per team). One group in each condition evaluated one office system, whereas the second group evaluated a second office system.

Concerning the study by Karat et al., Gray and Salzman [5] summarizes:

“This study handled most of the threats to internal validity well ... The mixed nature of the groups limits the generalization of their findings. ... The main failing of this study was with statistical conclusion validity. Few statistical tests were reported, and those that were reported failed to control for the Wildcard effect (Remark by Frøkjær and Larusdottir: Here “Wildcards” are participants who are significantly better or worse than average and whose performance in the conditions of the study do not reflect the UEM, but their Wildcard status.) Hence, although the results regarding the superiority of user testing to walkthroughs may be interesting and suggestive, they may not be generalizable beyond this study’s testing conditions.”

A direct comparison of the Karat et al. study with the present one is impossible because: (a) the walkthrough method used by Karat et al., which is combining techniques known from both cognitive walkthrough and heuristic evaluation, is quite different from both CW and HE used in our study; (b) the participants’ professional backgrounds are different from those of the evaluators in our study.

Just one of the important results found by Karat et al are mentioned, namely that team walkthroughs were more effective than individual walkthroughs. This supports the idea used in our study of taking reports from groups of evaluators as the unit of measurement instead of reports from individual evaluators.

Discussion Related to Cuomo and Bowen (1994)

Cuomo and Bowen [2] compare the three inspection-based evaluation methods CW, HE, and the Smith and Mosier guidelines. Their objective was to learn more about what types of usability

The study by Cuomo and Bowen was based on results from only one evaluation experiment with each method. We had experiments from five or six groups by each method or method combination. Another important difference is that Cuomo and Bowen's evaluators were highly educated and specialized human factors professionals with a solid knowledge also about the domain area of the system evaluated.

Using Norman's seven stage model of user activity, Cuomo and Bowen were able to show that the CW almost exclusively identified issues within the action specification stage, while guidelines and HE cover more of the stages. We have not tried any detailed analysis of this kind, but our results seem to support their main conclusions. Thus, Table 4 and 5 show that HE has induced the evaluators to take a wider view of the usability problems. When HE and TA tests are combined the coverage is the widest.

Cuomo and Bowen also conclude that CW was best at predicting problems that cause users noticeable difficulty as observed during a usability study. But they have no statistical support for this conclusion, and their data supporting this point actually show very small differences between CW and HE; and our results differ at this point. Table 6 and 7 present how the highly critical and the severe problems, i.e. the problems we consider to impede users have been uncovered and properly graded much better by the groups doing HE or TA than the groups doing CW.

Discussion Related to John and Marks (1997)

John and Marks [10] present a case study that tracks usability problems predicted with six UEMs, namely. claims analysis, CW, GOMS, HE, user action notation, and simply reading the specification. The predictive power of each UEM is assessed by comparing the predictions given by it to the results of user tests. They also measure what they call (a) the persuasive power of a method, i.e. a measure of the number of identified problems that led to changes in the implemented code, and (b) the design-change effectiveness which gives information about how the implemented changes may reduce the number of problems users experience, leave performance the same, or introduce more problems than before.

John and Marks's results are based upon only one evaluator using each of the methods; the involved evaluators are quite different in their educational and professional backgrounds; and the evaluations are based upon written specifications, not a prototype or a running version of the evaluated system. Although this study proposes interesting new concepts of importance for more adequate comparisons of UEMs, the "lessons learned" reported seem to us very uncertain and not relevant in comparison with ours. (See also John Carroll's critique of the study by John and Marks which are commented in [9].)

5 CONCLUSIONS

The effectiveness of predicting usability problems with HE and TA was found to be significantly higher than that of CW. A usability experiment performed by a group of three evaluators using HE or TA on an average uncovered approximately 80% more usability problems than a group using CW. The strict GOMS-like walkthrough built into CW seems to be unrealistically narrow focused compared to the users' actual activity during system interaction.

HE and TA were found to be complementary in the sense that they revealed somewhat different problems. We have found that one usability experiment using TA after a HE experiment on an average reveals about 25% of all the usability problems, while TA used after CW reveals only 15%. But this difference is not statistically significant because of quite large variances among

the six evaluator groups trying this method combination. The individual differences among evaluators seem to have important effects, even in our group-based evaluations.

The combination CW and TA has lesser predictive power than HE-TA. This might not be too surprising as both CW and TA are based upon the evaluator's task scenarios; and these task scenarios can not be expected to cover all aspects of the evaluated system when the system is more complex than walk-up and use systems.

All methods are well capable of uncovering the Highly Critical Problems (HCP). But reaching a good coverage of the Severe Problems (SP) by just one of the methods seems impossible to an average group of inexperienced evaluators. The combination, first HE and then TA, improves the coverage of Severe Problems from 25% to 36%. Moreover, this combination eliminates one of the most important weaknesses of the HE method when used by non-experts, the proneness to addressing many false problems.

In brief summary, this study shows that the idea of combining HE and TA, as brought forward in the literature [1: p. 87], effectively improves the methods suitability to uncover the important problems of smaller interactive software—without disturbing the picture by false problems, or over-graded Cosmetic Problems (CP).

ACKNOWLEDGMENTS

We thank Heather Desurvire, Morten Hertzum, Jakob Nielsen and John Rieman for their highly constructive critique of a preliminary design of this evaluation project, giving us many judicious proposals for improvements. We are exceptionally grateful to Rolf Molich who not only offered his advise during the preparation of the project, but continued to support us with his experience and ideas during the accomplishment of the study. Niels Bülow Andersen, Jette Holm Broløs, Morten Hertzum Kasper Hornbæk, Peter Naur, Ketil Perstrup and Kristian Bang Pilgaard read an earlier version of the paper and gave us numerous indispensable proposals for clarifying the presentation. Finally, we owe thanks to all the students who decided to participate in this research project by offering their exertion and enthusiasm as evaluators.

REFERENCES

- 1 Baecker, R.M., Grudin, J., Buxton, W.A.S., and Greenberg, S. [Authors and eds.] (1995) *Readings in Human-Computer Interaction: Toward the Year 2000*, Second edition, Morgan Kaufmann Publishers, Inc., San Francisco, California.
- 2 Cuomo, D.L., and Bowen, C.D. (1994) Understanding usability issues addressed by three user-system interface evaluation techniques. *Interacting with Computers*, 6, 1, 86-108.
- 3 Desurvire, H. W., Kondziela, J. M., and Atwood, M. E (1992) What is gained and lost using evaluation methods other than empirical testing. In Monk, A., Diaper, D., and Harrison, M. D. (eds.) *People and Computers VII*, Cambridge University Press, Cambridge, U. K.
- 4 Dix, A., Finlay, J., Abowd, G., and Beale, R. (1993) *Human-Computer Interaction*, Prentice-Hall, Englewood Cliffs, N. J.
- 5 Gray, W. D., and Salzman, M C. (1998) Damaged Merchandise? A Review of Experiments That Compare Usability Methods, *Human-Computer Interaction*, Vol. 13, pp. 203-261.
- 6 Hertzum, M., and Frøkjær E. (1996) Browsing and Querying in Online Documentation: A Study of User Interfaces and the Interaction Process, *ACM Transactions on Computer-Human Interaction*, 3, 2, pp 136-161.
- 7 Jacobsen, N. E., Hertzum, M., and John, B.E. (1997) The Evaluator Effect in Usability Tests, *Proc. ACM CHI '98*, Late-Breaking Results, pp. 255-256.
- 8 Jeffries, R., Miller, J. R., Wharton, C., and Uyeda, K. M. (1991) User interface evaluation in the real world: A comparison of four techniques, *Proc. ACM CHI '91 Conference*. (New Orleans, LA, 28 April - 2 May) ACM, NY.
- 9 John, B. E. (1998) On our case study of claims analysis and other usability evaluation methods, *Behaviour & Information Technology*, Vol. 17, 4, 244-246.
- 10 John, B. E., and Marks, S. J. (1997) Tracking the effectiveness of usability evaluation methods, *Behaviour & Information Technology*, Vol. 16, 4/5, 188-202.
- 11 John, B. E., and Mashyna, M. M. (1997) Evaluating a Multimedia Authoring Tool, *Journal of the American Society for Information Science*, 48(11):1004-1022.
- 12 Jørgensen, A. H. (1990) Thinking-aloud in user interface design: a method promoting cognitive ergonomics, *Ergonomics*, Vol. 33, pp. 501-507.
- 13 Karat, C. (1994) A comparison of user interface evaluation methods. In J. Nielsen and R.L. Mack (eds.) *Usability Inspection Methods*. New York: John Wiley, 1994.
- 14 Karat, C., Campbell, R., and Fiegel, T. (1992) Comparison of empirical testing and walkthrough methods in user interface evaluation, *Proc. ACM CHI '92 Conference*, (Monterey, California, 3 - 7 May).
- 15 Molich, R. (ed.), Beyer, P., Carstensen, P., Jørgensen, A. H., and Pedersen, F. H. (1986) *Brugervenlige edb-systemer*, Teknisk Forlag, København. In Danish. New edition: Molich, R. (1994)) *Brugervenlige edb-systemer*, Teknisk Forlag, København.
- 16 Nielsen, J. (1992) Evaluating the thinking-aloud technique for use by computer scientists, In Hartison, H. R., and Hix, D. (Eds.), *Advances in Human-Computer Interaction*, Vol. 3, Ablex. Norwood, N. J.
- 17 Nielsen, J. (1992) Finding usability problems through heuristic evaluation, *Proc. ACM CHI '92 Conference*, (Monterey, California, 3 - 7 May).
- 18 Nielsen, J. (1993) *Usability Engineering*, Academic Press, San Diego.
- 19 Nielsen, J., and Mack, R.L. [Eds.] (1994) *Usability Inspection Methods*. New York: John Wiley.
- 20 Nielsen, J., and Molich, R. (1990): Heuristic evaluation of user interfaces, *Proc. ACM CHI '90 Conference*, (Seattle, WA, 1 - 5 April).
- 21 Nielsen, J., and Philips, V. L. (1993) Estimating the relative usability of two interfaces: Heuristic, formal, and empirical methods compared. *Proc. ACM InterCHI'93 Conf*. (Amsterdam, The Netherlands, 24-29 April).
- 22 Polson, P., Lewis, C., Rieman, J., and Wharton, C. (1992) Cognitive walkthroughs: A method for theory-based evaluation of user interfaces, *International Journal of Man-Machine Studies*, 36.
- 23 Shneiderman, B. (1992) *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 2. edition, Addison-Wesley, Reading, MA.
- 24 Weinstein, S. (1992) *The Elm Mail System: A Replacement Mailer for all Unix Systems, Use Net Version (2.4 Release System)*, DIKU, København.
- 25 Wharton, C., Rieman, J., Lewis, C., and Polson, P. (1994) The cognitive walkthrough method: A practitioner's guide. Published in Nielsen, J. and Mack, R. L. (eds.) *Usability Inspection Methods*, Wiley, New York.