*International Journal of Human-Computer Interaction*, vol. 15, no. 1 (2003), pp. 183-204.
Preprint version. Published version at: https://dx.doi.org/DOI:10.1207/S15327590IJHC1501_14

# The Evaluator Effect: A Chilling Fact about Usability Evaluation Methods

Morten Hertzum[1] and Niels Ebbe Jacobsen[2]

[1] Centre for Human-Machine Interaction, Risø National Laboratory, Denmark

[2] Nokia Mobile Phones, Denmark

**Abstract**. Computer professionals have a need for robust, easy-to-use usability evaluation methods (UEMs) to help them systematically improve the usability of computer artefacts. However, cognitive walkthrough, heuristic evaluation, and thinking-aloud studies – three of the most widely used UEMs – suffer from a substantial evaluator effect in that multiple evaluators evaluating the same interface with the same UEM detect markedly different sets of problems. A review of eleven studies of these three UEMs reveals that the evaluator effect exists for both novice and experienced evaluators, for both cosmetic and severe problems, for both problem detection and severity assessment, and for evaluations of both simple and complex systems. The average agreement between any two evaluators who have evaluated the same system using the same UEM ranges from 5% to 65%, and no one of the three UEMs is consistently better than the others. While evaluator effects of this magnitude may not be surprising for a UEM as informal as heuristic evaluation, it is certainly notable that a substantial evaluator effect persists for evaluators who apply the strict procedure of cognitive walkthrough or observe users thinking out loud. Hence, it is highly questionable to use a thinking-aloud study with one evaluator as an authoritative statement about what problems an interface contains. Generally, the application of the UEMs is characterised by (1) vague goal analyses leading to variability in the task scenarios, (2) vague evaluation procedures leading to anchoring, and/or (3) vague problem criteria leading to anything being accepted as a usability problem. The simplest way of coping with the evaluator effect, which cannot be completely eliminated, is to involve multiple evaluators in usability evaluations.

**Keywords**: Usability evaluation methods, reliability, evaluator effect, cognitive walkthrough, heuristic evaluation, thinking-aloud studies.

## 1 Introduction

Computer professionals need robust, easy-to-use usability evaluation methods (UEMs). This study is about three prominent UEMs: cognitive walkthrough (CW), heuristic evaluation (HE), and thinking-aloud study (TA). CW was introduced by Lewis, Polson, Wharton, and Rieman (1990) and consists of a step-by-step procedure for evaluating the action sequences required to solve tasks with a system. HE was introduced by Nielsen and Molich (1990) and is an informal inspection technique with which evaluators test the system against a small number of interface heuristics. TA was introduced in systems development around 1980 (Lewis, 1982) and is probably the single-most important method for practical evaluation of user interfaces (Nielsen, 1993). These three UEMs span large differences in their approach to usability evaluation but share the common goal of supporting systems developers or usability specialists in identifying the parts of a system that cause users trouble, slow them down, or fit badly with their preferred ways of working, commonly termed usability problems.

This study is about whether evaluators who evaluate the same system with the same UEM detect – roughly – the same problems in the system. This issue is frequently neglected in UEM research as well as in practice, probably due to lack of awareness of the magnitude of the disagreements combined with prioritising swift and useful results over reliability and completeness. Several studies provide evidence that evaluators using the same UEM detect markedly different sets of usability problems when they evaluate a system. Evaluators also seem to

differ substantially in their rating of the severity of the detected problems. In the following, we will term differences in evaluators' problem detection and severity ratings the *evaluator effect*. We are well aware that a low evaluator effect is only one desirable property of a UEM. We specifically want to emphasise the distinction between reliability (i.e., the extent to which independent evaluations produce the same result) and validity (i.e., the extent to which the problems detected during an evaluation are also those that show up during real-world use of the system). The evaluator effect is a measure of reliability only. To our knowledge, the validity of UEMs has not been investigated.

This study brings together the results of previous studies and newly derived results from studies that contain the data necessary to investigate the evaluator effect but did not address this issue. With these data, we intend to show that the evaluator effect cannot be dismissed as a chance incident, an artefact of the peculiarities of a single study, or a weakness of a particular UEM. Notably, the evaluator effect is also of concern to TA, which is generally considered the most authoritative method for identifying usability problems. By looking at differences and commonalities of the reviewed studies and UEMs, we then discuss where the UEMs fall short of providing evaluators with the guidance necessary to perform reliable usability evaluations. This leads to input for improving current UEMs but also to the realisation that the evaluator effect will to a considerable extent have to be managed rather than eliminated.

The next section provides a brief introduction to the three UEMs. In section 3, two measures of the evaluator effect are defined and discussed. In section 4, we review eleven UEM studies that provide empirical data on the evaluator effect. As examples, three of these studies are described in more detail. In section 5, we discuss possible causes for the evaluator effect. Finally, the concluding section aims at advising practitioners on how to cope with the evaluator effect in UEMs.

# 2 A brief introduction to the methods

The following descriptions of CW, HE, and TA are mere introductions provided to give a flavour of how usability evaluation is approached with the three methods. Guidance on how to conduct evaluations with CW, HE, and TA can be found in Wharton, Rieman, Lewis, and Polson (1994), Nielsen (1994a), and Dumas and Redish (1993), respectively.

## 2.1 Cognitive walkthrough (CW)

CW (Lewis et al., 1990; Lewis & Wharton, 1997; Polson, Lewis, Rieman, & Wharton, 1992; Wharton, Bradford, Jeffries, & Franzke, 1992; Wharton et al., 1994) has been devised to enable computer professionals to detect usability problems in a user interface based on a detailed specification document, screen mock-ups, or a running system. CW is particularly suited to evaluate designs before testing with users becomes feasible and as a supplement to user testing in situations where users are difficult or expensive to recruit. Also, CW was initially developed for evaluating walk-up-and-use systems, although it has later been applied to more complex interfaces. It has been recommended that CW should be performed by groups of co-operating evaluators but the descriptions of the method maintain that it can also be performed by evaluators working individually. CW is based on a cognitive theory of exploratory learning called CE+ (Polson & Lewis, 1990; Polson et al., 1992), and a basic understanding of this theory is a definite advantage when performing a walkthrough.

The procedure for CW consists of a preparation phase and an execution phase. In the preparation phase the evaluator describes a typical user, chooses the tasks to be evaluated, and constructs a correct action sequence for each task. When this is done, the execution phase can begin. For each action in the action sequences the evaluator asks four questions[1]: (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 with the effect trying to be achieved? (4) If the correct action is performed, will the user see that progress is being made toward solution of the task? With the description of the user in mind the evaluator decides whether each question leads to success or failure. In case of failure a usability problem has been detected. After all actions have been evaluated, the CW is completed by merging the detected problems into one non-duplicate list.

## 2.2 Heuristic evaluation (HE)

HE (Nielsen & Molich, 1990; Nielsen, 1992; 1993; 1994a) is an informal UEM that enables evaluators to detect usability problems in an interface based on screen mock-ups or a running system. The informality has implications for the reliability and coverage of heuristic evaluations but is considered necessary to get computer

professionals to adopt the method. Any computer professional should be able to apply HE but the informality of the method leaves much to the evaluator. Consequently, the evaluator's skills and expertise have a large bearing on the results. A single inexperienced evaluator is unlikely to produce sufficiently good results. For this reason HE prescribes that a small group of evaluators individually inspect the system. In addition, Nielsen (1992) has found that the effectiveness of HE can be substantially improved by having usability specialists as evaluators.

The procedure for HE involves having a small group of evaluators examine an interface and judge its compliance with a small set of recognised usability principles – the heuristics. Nielsen (1994a) provides a set of ten general heuristics, which state that the system should[2]: (1) provide visibility of system status, (2) ensure match between system and the real world, (3) allow for user control and freedom, (4) be consistent and follow standards, (5) prevent errors, (6) utilise recognition rather than recall, (7) allow for flexibility and efficiency of use, (8) provide aesthetic and minimalist design, (9) help users recognise, diagnose, and recover from errors, and (10) provide help and documentation. Each evaluator goes through the interface and inspects the various dialogue elements and compares them with the heuristics. In addition to the checklist of general heuristics to be considered for all dialogue elements, the evaluator may also consider any additional usability principles or results that seem to be relevant for any specific interface element. To ensure independent and unbiased evaluations, the evaluators are only allowed to communicate and aggregate the results of their evaluations after they have completed their own, individual inspection of the interface.

### 2.3 Thinking-aloud study (TA)

Since TA was first introduced in systems development numerous variations of the method have been employed and today there is no definitive definition of the aim and usage of the method, and no single accepted procedure to follow. TA is used in various situations, with various goals, and both early and late in the development cycle (see, Nielsen, 1994b; Dumas & Redish, 1993). TA can, for example, be performed by usability specialists in a usability laboratory with video cameras and one-way mirrors or it can be performed in the field and analysed on-the-fly by systems developers. The common core of TA is that it involves a small number of users who think out loud while solving tasks using the system that is being tested and an evaluator who detects usability problems by observing the users and listening in on their thoughts. It is generally held that at least 4-5 users are necessary to detect the majority of the problems in a system, but the necessary number of users varies considerably with the aim of the test and the complexity and quality of the system (see Lewis, 1994).

The general procedure for TA consists of a preparation phase followed by a number of test sessions, normally one for each user. In the preparation phase the people conducting the test familiarise themselves with the work environment where the system is going to be used, define appropriate tasks, and recruit users. The test sessions are administered by a facilitator, who may at the same time be the person evaluating when the users experience problems. Each session consists of an introduction to familiarise the user with the test situation, the actual test, and a debriefing of the user. In the introduction, the facilitator should teach the user to think out loud since this is an unnatural thing to do for users and experience indicates that without teaching – and some encouragement during the session – only few users are capable of giving valuable verbal reports about their work. The actual test is initiated by reading the first task out loud, and handing it over to the user who solves it while thinking out loud. After finishing the first task, the second is presented in a similar manner, and so forth. When the user has finished all tasks or time runs out, the user is debriefed to provide any additional insights into the system and to relax after the test session. After all test sessions have been run, the evaluator produces a complete, non-duplicate list of the detected problems.

## 3 Measuring the evaluator effect

Previous studies (for example, Hertzum & Jacobsen, 1999; Nielsen, 1992; Jacobsen, Hertzum, & John, 1998) have used the average detection rate of a single evaluator as their basic measure of the evaluator effect. This measure relates the evaluators' individual performance to their collective performance by dividing the average number of problems detected by a single evaluator by the number of problems detected collectively by all the evaluators. These calculations are based on unique problems; that is, after duplicate problems within and between evaluators have been eliminated, see Equation 1.

$$\text{Detection rate} = \text{Average of } \frac{|P_i|}{|P_{all}|} \text{ over all n evaluators} \quad (1)$$

In the equation, $P_i$ is the set of problems detected by evaluator i (i.e., the problem list of evaluator i) and $P_{all}$ is the set of problems detected collectively by all n evaluators. The detection rate is easy to calculate and it is available for the eleven studies reviewed in this paper. However, the detection rate suffers from two drawbacks. First, the lower bound for the detection rate varies with the number of evaluators. Using only one evaluator the detection rate will always be 100%, using two evaluators it will be at least 50% (when there is no overlap between the two evaluators' problem lists), and using n evaluators it will be at least 100/n percent. When the number of evaluators is small it is important to interpret the detection rate as a value between the lower bound and 100%, not between 0% and 100%. Otherwise, the detection rate will appear higher than it actually is. Second, and related, the detection rate rests on the assumption that the number of problems found collectively by the evaluators is identical to the total number of problems in the interface. A small group of evaluators is, however, likely to miss some problems and then the detection rate becomes overly high. Adding more evaluators will normally lead to the detection of some hitherto missed problems, and this improvement of the evaluators' collective performance is reflected in the detection rate as a decrease in the performance of individual evaluators.

To avoid the problems caused by relating the performance of single evaluators to the collective performance of all evaluators, the any-two agreement measures to what extent pairs of evaluators agree on what problems the system contains. The any-two agreement is the number of problems two evaluators have in common divided by the number of problems they collectively detect, averaged over all possible pairs of two evaluators, see Equation 2.

$$\text{Any-two agreement} = \text{Average of } \frac{|P_i \cap P_j|}{|P_i \cup P_j|} \text{ over all } \tfrac{1}{2}n(n-1) \text{ pairs of evaluators} \quad (2)$$

In the equation, $P_i$ and $P_j$ are the sets of problems detected by evaluator i and evaluator j and n is the number of evaluators. The any-two agreement ranges from 0%, if no two evaluators have any problem in common, to 100%, if all evaluators have arrived at the same set of problems. It should be noted that the any-two agreement measures agreement only. A high any-two agreement is no guarantee that all, or even most, problems in the interface have been detected. The any-two agreement is our preferred measure of the evaluator effect[3] but we have the data necessary to calculate it for only a subset of the studies reviewed in the next section.

# 4 Studies of the evaluator effect in CW, HE, and TA

Reviewing the UEM literature we found six studies that explicitly aimed at investigating the evaluator effect (Hertzum & Jacobsen, 1999; Jacobsen & John, 1999; Jacobsen et al., 1998; Nielsen, 1992; 1994a; Nielsen & Molich, 1990). Two additional studies touched on the evaluator effect in a broader sense (Molich et al., 1998; 1999). In addition to that, one study, which did not particularly aim at investigating the evaluator effect, contained data that enabled us to investigate it (Lewis et al., 1990). Finally, the authors of two UEM studies generously provided us with additional data that enabled us to investigate the evaluator effect in their studies (Connell & Hammond, 1999; Dutt, Johnson, & Johnson, 1994).

We would have liked our review to include studies where CW was performed by groups of co-operating evaluators because this has been suggested as an improvement of the method (Wharton et al., 1992; 1994). We would also have welcomed studies where HE was performed by evaluators who aggregated the results of their individual inspections to a group output because this is how HE is described. We have, however, only been able to find studies where CW and HE are performed by evaluators working individually. It is currently unknown what effect collaboration among evaluators has on the reliability of usability evaluations. For software inspections it has been found that having the inspectors meet to aggregate the results of their individual inspections leads to the detection of few new defects but the loss of a number of defects that were originally detected by individual inspectors – meeting losses outweigh meeting gains (Miller, Wood, & Roper, 1998).

## 4.1 Overview

Table 1 shows that substantial evaluator effects have been found for all three UEMs across a range of experimental settings. Specifically, the evaluator effect is neither restricted to novice evaluators nor to evaluators knowledgeable of usability in general. The evaluator effect is also found for evaluators with experience in the specific UEM they have been using (Jacobsen et al., 1998; Lewis et al., 1990; Molich et al., 1998; 1999). Furthermore, the evaluator effect is not affected much by restricting the set of problems to only the severe problems. The first three columns in Table 1 contain information on which study is reviewed, the UEM investigated, and the type of system evaluated. The fourth column tells whether the system was tested against set tasks, which may have an impact on the evaluator effect because they tend to make the test sessions more similar. The fifth column gives the total number of unique problems detected by the evaluators collectively. Duplicated problems have been eliminated from this figure; that is, when an evaluator has detected the same problem twice or when two evaluators have detected the same problem, this problem is only counted once. The sixth column gives the number and profile of the evaluators. The seventh column gives the detection rate (see Equation 1). For example, in the study by Lewis et al. (1990), an evaluator, on average, detected 65% of the total number of problems detected collectively by the four evaluators. In some but not all studies, a set of severe problems was extracted from the total set of problems. The eighth column gives the detection rate for severe problems only, as it is interesting to know whether the evaluator effect is smaller for severe problems than for all problems. For example, in the study by Hertzum and Jacobsen (1999), an evaluator detected, on average, 21% of the severe problems, which was only slightly more than when calculating the detection rate for the full set of known problems. The last column gives the any-two agreement (see Equation 2) for the studies where we have been able to calculate it. For example, in the study by Connell and Hammond (1999), the average agreement between any two evaluators is 5%-9%.

Table 1 appears at the end of the paper

As described in Section 3, a study with few evaluators is likely to yield an overly high detection rate because some problems remain unnoticed by all the evaluators. This leads us to assume that adding more evaluators to the studies that originally involved only a few evaluators will cause a drop in the achieved detection rates (see, Lewis, this issue, for a formula for estimating the drop in detection rate based on data from the first few evaluators). Thus, an overall estimate of the detection rate should probably lean toward the studies with the larger number of evaluators, and these studies generally report the lower detection rates.

## 4.2 A closer look at the reviewed studies

To provide the detail necessary to assess the credibility of the reviewed studies, this section points out the special characteristics of the individual studies and describes three studies, one for each UEM, in more detail.

**Cognitive walkthrough (CW)**

Lewis et al. (1990) had four evaluators individually perform a CW and found that the evaluators were fairly consistent and that they collectively detected almost 50% of the problems revealed by an empirical evaluation with 15 users. The generalisability of these results is difficult to assess, however, because three of the evaluators were creators of CW and had discussed the general trends of the empirical evaluation prior to their walkthroughs (Lewis et al., 1990, pp. 238-239). Dutt et al. (1994) had three evaluators individually perform a CW and they are unusually consistent. The authors note that "the number of problems found is relatively low given the quality of the interface." This could indicate that the evaluators did not find all the problems in the interface or that they applied a rather high threshold for the amount of difficulty or inconvenience inflicted on the user before they reported a problem. The studies of Lewis et al. (1990) and Dutt et al. (1994) are based on the first version of CW, which used a single-page form with nine general questions and several sub-questions to evaluate each action. Jacobsen and John (1999) studied two novice evaluators as they learned and used the simpler current version of CW. Based on a detailed system specification document the evaluators individually selected the tasks to be evaluated and spent 22-25 hours evaluating the system. The agreement between the evaluators was disappointingly low with only 6% of the problems being detected by both evaluators.

Hertzum and Jacobsen (1999) had 11 first-time users of CW evaluate a Web-based library system against three set tasks. Thus, the study bypassed task selection to focus on the construction and walkthrough of the action sequences. The evaluators, who were graduate students in computer science, received two hours of

instruction in the CW technique. This instruction consisted of a presentation of the practitioner's guide to CW (Wharton et al., 1994), a discussion of this guide, and an exercise where the evaluators got some hands-on experience and instant feedback. As a rough estimate, each evaluator spent 2-3 hours individually completing his CW, which had to be documented in a problem list describing each detected problem and the CW question that uncovered it. Based on the eleven problem lists, the two authors of Hertzum and Jacobsen (1999) independently constructed a master list of unique problem tokens. The authors agreed on 80% of the problem tokens and resolved the rest through discussion. The evaluators differed substantially with respect to which and how many problems they detected. The largest number of problems detected by a single evaluator was 13, whereas the lowest was two. As much as 58% of the 33 problems detected collectively by the evaluators were only detected once, and no single problem was detected by all evaluators.

**Heuristic evaluation (HE)**

Nielsen and Molich (1990) report on the heuristic evaluation of four simple walk-up-and-use systems. Two of the systems (Savings and Transport) were running versions of voice-response systems, the two other systems (Teledata and Mantel) were evaluated on the basis of screen dumps. In all four evaluations it was found that aggregating the findings of several evaluators had a drastic effect in the interval from one to about five evaluators. After that the effect of using an extra evaluator decreased rapidly and seemed to reach the point of diminishing returns at aggregates of about ten evaluators. In a second study, Nielsen (1992) compared three groups of evaluators who performed a HE of a simple system giving people access to their bank accounts. The three groups of evaluators were novices, regular usability specialists, and specialists in both voice-response systems and usability (the double specialists). The performance of the evaluators, who made their evaluation from a printed dialogue that had been recorded from the system, increased with their expertise. However, even the double specialists displayed a notable evaluator effect, and they differed just as much in their detection of severe problems as they did for problems in general. In a third study, Nielsen (1994a) reports on a HE of a prototype of a rather complex telephone company application intended for a specialised user population. This study provides evidence that heuristic evaluations of more complex systems are also subject to a substantial evaluator effect.

Connell and Hammond (1999) conducted two experiments to investigate the effect of using different sets of usability principles in usability evaluations. We focus on the evaluators using the ten HE heuristics as their usability principles. In the first experiment, a group of novice evaluators and a group of evaluators with HCI knowledge applied HE to a hypermedia browser. Problems were collected by observing the evaluators who were asked to think out loud. In the second experiment, a group of novice evaluators applied HE to an interactive teaching system. Here, the normal HE procedure was followed in that the evaluators reported their findings themselves. The detection rates in both experiments were among the lowest obtained in the studies of HE. Connell and Hammond (1999) argue that they have been more cautious not to group distinct problems into the same unique problem token and therefore get lower detection rates. The duplicate-elimination process where the problem lists of individual evaluators are merged into one master list of unique problem tokens can result in misleadingly high detection rates if the problems are grouped into too few, overly broad problems. Another explanation could be that the number of problems in the interface increases with the complexity of the system, and this reduces the likelihood that two evaluators will detect the same set of problems. If that is the case, studies of the evaluator effect should be performed on realistically complex systems, such as those used by Connell and Hammond (1999).

**Thinking-aloud study (TA)**

There has not been much focus on the evaluator effect in TA. In two independent studies Molich et al. (1998; 1999) have investigated to what extent usability laboratories around the world detect the same problems in a system based on a TA. There are, inarguably, more differences between laboratories than the evaluators performing the evaluations (e.g., differences in test procedure and different individuals participating as users). We have included these two studies in the review but want to emphasise that they differ from the other reviewed studies – lower agreement must be expected because more constituents of the evaluation were allowed to vary. In the first study (Molich et al., 1998), three[4] commercial usability laboratories evaluated the same running system with TA based on a two-page description of the primary user group and the aim of the test. As much as 129 of the 141 reported problems were only detected once. In the second study (Molich et al., 1999), six[5] usability laboratories evaluated a commercial web-based system with TA, this time based on a more precise description of the users and the aim of the test. Again, the laboratories disagreed substantially in that 147 of the 186 reported problems were only detected once.

Only Jacobsen et al. (1998) have aimed specifically at revealing the evaluator effect in TA. In this study, four HCI researchers – two with extensive TA experience and two with some TA experience – independently analysed the same set of videotapes of four usability test sessions. Each session involved a user thinking out loud while solving set tasks in a multimedia authoring system. The evaluators, who also had access to the system and its specification document, were asked to report all problems appearing in the four videotaped test sessions. The evaluators were not restricted in the time they spent analysing the videotapes, but to minimise individual differences in their conceptions of what constitutes a usability problem, nine set criteria were used. Hence, the evaluators were requested to detect problems according to the nine criteria, and they were asked to report time-stamped evidence and a free-form description for each problem. Based on the evaluators' problem lists, two of the authors of Jacobsen et al. (1998) independently constructed a master list of unique problem tokens. They agreed on 86% of the problem tokens, and by discussing their disagreements and the problems they did not share, a consensus was reached. As much as 46% of the problems were only detected by a single evaluator, and another 20% by only two evaluators. Compared to the two less experienced evaluators, the two evaluators with extensive TA experience spent more time analysing the videotapes and found more problems. The average detection rate for the two experienced evaluators was 59% but they agreed on only 40% of the problems they collectively detected. The average detection rate for the two less experienced evaluators was 45% and they agreed on 39% of the problems they collectively detected. Even though the study was set up to minimise the evaluator effect by being more restrictive than most practical TA studies, a substantial evaluator effect remained.

## 4.3 Severity judgements

Seven studies, corresponding to 14 experiments, include an assessment of problem severity. In the study by Connell & Hammond (1999) problem severity was assessed on a seven-point rating scale and severe problems were defined as those receiving one of the three highest rates. The other studies that assessed problem severity did so by dividing the problems into two categories: severe and non-severe. This was done by having the evaluators point out the problems that ought to be fixed before release of the system (Hertzum & Jacobsen, 1999), by having the evaluators point out the ten most severe problems (Jacobsen et al., 1998), by stipulating a set of core problems (Molich et al., 1999), or based on expected impact on the user (Nielsen & Molich, 1990; Nielsen, 1992; 1994a). Four experiments display an appreciably higher detection rate for severe problems; the other ten experiments display largely no difference between the detection rate for all problems and the detection rate for severe problems only (see Table 1). Thus, the evaluator effect is not merely a disagreement about cosmetic, low-severity problems, which are more or less a matter of taste. For all three UEMs, a single evaluator is unlikely to detect the majority of the severe problems that are detected collectively by a group of evaluators.

Another way of looking at the evaluator effect is to investigate to what extent evaluators agree on what constitutes a severe problem. In a number of the reviewed studies several of the evaluators were also asked to judge the severity of the problems on the complete list of unique problems. The evaluators' assessments of problem severity are suitable for comparison because they are made independently and based on the same list of problems. The evaluators could however be biased toward perceiving the problems they originally detected themselves as more severe than the problems they missed. Lesaigle and Biers (2000) report a statistically significant bias for four of the 13 evaluators who assessed problem severity in their study. Jacobsen et al. (1998) and Nielsen (1994a) have also investigated this potential bias and found that it was negligible.

The evaluators in Jacobsen et al. (1998) received the complete list of unique problems with a short description of each unique problem and additional information about, among other things, the number of users experiencing it and the number of evaluators detecting it. Each evaluator was presented with a scenario in which a project manager had constrained the evaluators to point out the ten most severe problems, as a tight deadline left room for fixing only those few problems in the next release. After they had created their top-10 lists, the evaluators were also asked to write down their strategy for creating their list. The strategies varied greatly among the evaluators and were based on multiple aspects such as the evaluators' favour for certain user groups, the number of evaluators and users encountering a problem, the violated problem criteria, expectations about real-world usage of the system, and so forth. All these aspects may catch important dimensions of problem severity but they also led the evaluators to select markedly different sets of problems for their top-10 lists.

Table 2, which covers one study for each of the three UEMs, shows the extent to which evaluators who assess problem severity agree on the set of severe problems. The table gives the any-two agreement among the evaluators with respect to which problems they considered severe and the average correlation between the severity ratings provided by any two evaluators. Nielsen (1994a) states that "the reliability of the severity ratings from single evaluators is so low that it would be advisable not to base any major investment of development time and effort on such single ratings." In Hertzum and Jacobsen (1999), 35% of the total set of severe problems were

only rated severe once. In Jacobsen et al. (1998), 56% of the problems on the top-10 lists were only rated severe once. Not a single problem was unanimously judged as severe in these two studies. In sum, Table 2 shows that the CW, HE, or TA performed by the evaluators did not give rise to a common agreement as to what constituted the central usability issues in the interfaces.

**Table 2**. Three studies where a group of evaluators judged problem severity. A dash ('-') indicates that the figure could not be calculated from the available data.

| Reference | UEM | Evaluated system | Evaluators who assessed severity | No. of severe problems | Any-two agreement on severity ratings | Average Spearman correlation (std. deviation) |
|---|---|---|---|---|---|---|
| Hertzum & Jacobsen, 1999 | CW | Web-based library | 6 CS graduate students | 20 | 28% | 0.31 (0.18) |
| Nielsen, 1994a | HE | Integrating | 11 usability specialists | - | - | 0.24 |
| Jacobsen et al., 1998 | TA | Multimedia authoring | 4 HCI researchers with TA experience | 25 | 20% | 0.23 (0.16) |

# 5 Discussion

The evaluator effect has been documented for different UEMs, for both simple and complex systems, for both paper prototypes and running systems, for both novice and experienced evaluators, for both cosmetic and severe problems, and for both problem detection and severity judgement. The question is not whether the evaluator effect exists, but why it exists and how it can be handled. We believe the principal reason for the evaluator effect is that usability evaluation involves interpretation. While some usability problems are virtually self-evident, most problems require that the evaluator exercises judgement in analysing the interaction between the users, their task, and the system. It should be noted that evaluator effects are not specific to usability evaluation. Inter-observer variability also exists for more matured cognitive activities such as document indexing (e.g., Funk, Reid, & McCoogan, 1983; Sievert & Andrews, 1991; Zunde & Dexter, 1969) and medical diagnosing (e.g., Corona et al., 1996; Cramer, 1997; Sørensen, Hirsch, Gazdar, & Olsen, 1993). In general, individual differences – often categorised into groups like cognitive abilities, expertise, motivation, personality, and skill acquisition – preclude that cognitive activities such as detecting and assessing usability problems are completely consistent across evaluators.

In analysing how interpretation enters into usability evaluations and gives rise to differences across evaluators, we have focused on where the UEMs fall short of providing evaluators with the guidance necessary for performing reliable evaluations. Three such shortcomings of the UEMs are discussed in the following: (1) vague goal analyses leading to the selection of different task scenarios, (2) vague evaluation procedures leading to anchoring, and (3) vague problem criteria leading to anything being accepted as a problem. In addition to discussing what causes the evaluator effect, we also make suggestions regarding how it can be dealt with.

## 5.1 Vague Goal Analyses

At least for complex systems, it is not practically possible to include all aspects of a system in one evaluation. Consequently, it is important to analyse what the evaluation is to achieve and focus it accordingly. Vague goal analysis prior to usability evaluation leaves many decisions about which aspects of the system to include in the evaluation to the evaluator's discretion. While evaluators may agree in general on the focus of an evaluation, small differences in their selection of which specific functions to evaluate for different system features may lead to considerable variability in the evaluators' final choice of evaluation tasks. Although evaluating different aspects of the system might not be thought of as an evaluator effect per se, it certainly impacts the results of a usability evaluation.

The outcome of the goal analysis can simply be a mental clarification but the goal analysis can also result in a set of task scenarios the system is to be evaluated against. HE relies on a merely mental clarification of the goal of the evaluation, CW makes use of task scenarios but includes no guidance on task selection, and the various versions of TA normally include task scenarios devised on the basis of interaction with target users. In five of the experiments reviewed in the previous section the evaluators received identical task scenarios for their evaluation;

in the other 13 experiments task scenarios were either not used at all or it was left to the individual evaluators to select them. In Jacobsen and John (1999), the two CW evaluators were to set up task scenarios by themselves, and 43% of the problems that were detected by one evaluator and missed by the other stemmed from tasks selected and evaluated by only one of the evaluators. The reviewed studies provide ample evidence that the evaluator effect is not eliminated by giving the evaluators identical task scenarios, but it must be suspected that vague goal analyses introduce additional variability.

Task selection, or the broader activity of goal analysis, seems a somewhat neglected aspect of the three reviewed UEMs. We suggest that to reduce the evaluator effect and in general improve the quality of their evaluations, evaluators should verify the coverage of their task scenarios in a systematic way. Such an analysis of task coverage is intended to ensure that all relevant system facilities are considered for inclusion in a task scenario and hence provides a basis for selecting the optimal subset of facilities for actual inclusion in the evaluation. The most important facilities to test will generally be the high-risk ones and those with a known or expected high frequency of use.

## 5.2 Vague evaluation procedures

Whereas TA and in particular CW provide the evaluator with a procedure describing the phases of the evaluation and how to complete them, HE does not offer much in terms of a procedure for driving the evaluation. The heuristics used in HE "seem to describe common properties of usable interfaces" (Nielsen, 1994a) but HE does not provide a systematic procedure for ensuring that all interface elements are evaluated against all heuristics. Thus, while the heuristics take one step toward pointing to the problems in the interface they still leave a considerable gap for the evaluator to close. We conjecture that the heuristics principally serve as a source of inspiration in that they support the evaluator in looking over the interface several times while focusing on different aspects and relations of it. The quality of this stream of new aspects and relations is that it leads the evaluator to consider still new questions about the usability of the interface. While novice evaluators may use the heuristics in this concurrent and inspirational way, experienced evaluators might chiefly get their inspiration from the experience they have accumulated during past evaluations. Thus, HE leaves room for using the heuristics in different ways and to different extents. This is a deliberate feature of HE, which is intended as an easily applicable informal method, but it also leads to an evaluator effect.

While a substantial evaluator effect may not be surprising for a UEM as informal as HE, it is certainly notable that the strict procedure of CW does not lead to consistently better agreement among the evaluators. In a CW, the evaluator specifies a user or a group of users (for example, "users with Mac experience"), which is the basis for answering the four questions for each action in the action sequences. It is a critical assumption of CW that posing the four questions helps evaluators reach reliable answers. It is, however, not evident to what extent this is the case. To answer the questions accurately, the evaluator needs to know quite a lot about how the specified users will react to different user interface properties and facilities – knowledge that is not typically made explicit in the user description (Hertzum & Jacobsen, 1999; Jacobsen & John, 1999). In case of insufficient knowledge of how the users will react to the interface, the walkthrough becomes inaccurate due to a phenomenon known as anchoring; that is, despite the evaluator's efforts the walkthrough ends up evaluating the system against a user who is much too similar to the evaluator to be representative of the actual users. Each of the four questions in CW drives evaluators to think of the user's behaviour in a certain situation but when the general user description becomes too fuzzy, the evaluators unintentionally substitute it with their own experience with the system. The anchoring hypothesis has been investigated by Jacobsen and John (1999), who kept track of the evaluators' learning and evaluation process through diaries written by the evaluators themselves. For both evaluators, the authors found examples of usability problems the evaluators reported in and credited to their CWs, although the evaluators had noticed these problems during their preparation phase up to 15 hours before encountering them as part of their walkthrough process. This illustrates that usability problems experienced personally by evaluators are likely to enter into their evaluations by showing up later as reported usability problems.

The anchoring hypothesis can readily be extended to HE, which is also an inspection method, but one could hope that it would not extend to TA where the evaluator observes, rather than imagines, target users interacting with the system. However, differences in TA evaluators' general views on usability, their personal experiences with the system under evaluation, their opinions about it, and so forth lead them to make some observations and remain blind toward others. We can only hypothesise that this is due to anchoring, but the magnitude of the resulting evaluator effect testifies to the considerable amount of interpretation involved in evaluating TA sessions, in spite of the rather concrete procedure.

The effect of adding more evaluators to a TA study resembles the effect of adding more users; both additions increase the overall number of problems found, and by comparable amounts. In fact, the study by Jacobsen et al. (1998) suggests that the geometric mean of the number of users and evaluators is a rule-of-thumb estimate of the total number of problems identified in a TA study (see Equation 3). This means that to maximise the number of problems found and, simultaneously, minimise the number of users and evaluators, the number of users and evaluators should be the same: Three evaluators individually observing three users are more productive in identifying usability problems than is one evaluator observing five users. It should be kept in mind that Equation 3 is derived from a study with only four users and four evaluators. This may not be enough to make reliable predictions for large numbers of users and evaluators.

$$\text{Number of problems found} \approx C \sqrt{\text{number of evaluators} \times \text{number of users}} \quad (3)$$

If we take as our premise that each evaluator examines the interface once per user, then setting the number of evaluators equal to the number of users maximises the number of examinations of the interface. Hence, the crucial factor to consider in deciding upon how many users and evaluators to involve in a TA study is the number of examinations of the interface. As the time invested and the hourly price for each evaluator is normally higher than for each user, it will probably not be cost-effective to have an equal number of users and evaluators but it could be considered to trade a couple of users for an extra evaluator. A positive side effect of this suggestion is that in comparing their results the evaluators will have an opportunity to discuss and learn from the nature and size of their disagreements, thus increasing their awareness of the evaluator effect.

Previous studies of how many users to include in TA studies have found that the number of problems detected can be modelled by the formula $N(1 - (1 - p)^u)$, where N is the total number of problems in the interface, p is the probability of finding the average problem when running a single, average user, and u is the number of users participating in the evaluation (Lewis, 1994; Nielsen & Landauer, 1993; Virzi, 1992). Both Equation 3 and the $N(1 - (1 - p)^u)$ formula predict diminishing returns for increasing numbers of users (and evaluators); that is, adding another user or evaluator will yield fewer and fewer hitherto unnoticed problems as the number of users and evaluators increases. However, Equation 3 rejects the idea of a total number of problems in the interface – rather the number of problems will keep increasing for each new user and evaluator. It should however be reemphasised that Equation 3 may not make reliable predictions for large numbers of users and evaluators.

TA studies with a constant number of evaluators yield diminishing returns for increasing numbers of users but Equation 3 indicates that whereas a TA study with one evaluator may close in on one value for the total number of problems in the interface, studies with more evaluators will close in on higher values. Thus, if formulas like $N(1 - (1 - p)^u)$ are used to estimate the total number of problems in an interface based on a TA study performed by a single evaluator we must expect that the number of problems is underestimated. Figure 1, from Jacobsen et al. (1998), depicts the number of problems detected as a function of the number of both evaluators and users. Each curve corresponds to a fixed number of evaluators. Looking at the evaluators' performance after they had analysed all four users, the average increase in problems found was 42% going from one to two evaluators, 20% going from two to three evaluators, and 13% going from three to four evaluators.

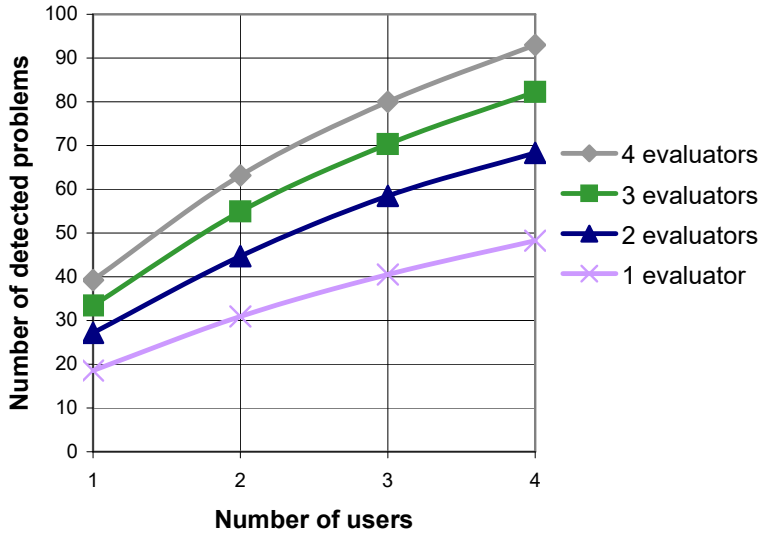


**Figure 1**. The number of problems detected by different numbers of users and evaluators in the TA study by Jacobsen et al. (1998). One evaluator analysing four users found on average 48 problems but collectively the four evaluators detected a total of 93 problems.

## 5.3 Vague problem criteria

Heuristics such as "ensure match between system and the real world" do not tell how big a mismatch is allowed to be before it becomes a problem. Similarly, CW provides no guidance on how quickly and effortlessly the user should notice that the correct action is available before this action must be said to be insufficiently noticeable. For TA it is also uncommon that the evaluators have explicit criteria defining when a difficulty or inconvenience experienced by the user constitutes a usability problem. However, in one of the reviewed studies of TA (Jacobsen et al., 1998), the evaluators were provided with nine predefined criteria defining when an observation should be recorded as a usability problem. Thus, differences in the evaluators' thresholds regarding when a difficulty or inconvenience becomes a problem are generally not regulated by the UEMs and must be suspected to contribute considerably to the evaluator effect. Evaluators are much more likely to disagree in their choice of threshold – and consequently on whether the difficulty or inconvenience inflicted on the user is sufficiently big to constitute a usability problem – than to hold downright contradictory opinions.

The second version of CW (Polson et al., 1992) made use of extensive criteria for supporting the evaluator in determining whether each of the questions led to the detection of a problem. This level of detail and explicitness was given up in the current version of the method as several studies described the second version of CW as prohibitively formal and tedious (Rowley & Rhoades, 1992; Wharton et al., 1992). In the current version of CW, however, the evaluator still repeats the same four questions for every action. For moderately complex systems, the number of actions may exceed a hundred; that is, more than hundred repetitions of the same question although for different actions. Along with the documentation of the walkthrough, this process indeed becomes tedious. In the study by Hertzum and Jacobsen (1999), several of the evaluators unintentionally skipped an action in the midst of an action sequence. Such slips are likely effects of the tedium of a process that requires the evaluators to meticulously follow a formal procedure. This leads to random differences among the evaluators, and illustrates that efforts to reduce the evaluator effect by providing more formal and complete problem criteria may prove ineffective. Instead of better performance the increased formality may introduce slips and other inconsistencies in the evaluators' behaviour.

Without a set of criteria defining what constitutes a usability problem, the reviewed studies end up accepting any problem report as a usability problem. In contrast, Nielsen (1993) distinguishes between usability problems (i.e., problems concerning *how* the system is to be operated) and utility problems (i.e., problems concerning *what* the system can do). As researchers, we need precise operational definitions of core concepts, such as usability problem, to make reliable studies of UEMs (see Gray & Salzman, 1998). Otherwise, two evaluators may make the same observations but report them differently due to differences in their understanding of what they are looking for. As practitioners, we are somewhat reluctant to adopt explicit problem criteria because they may

favour agreement among evaluators over detection of all problems of practical importance. That is, a shared understanding of what constitutes a usability problem may not only reduce the evaluator effect but also cause evaluators to systematically miss certain types of problems. We believe explicit problem criteria can reduce the evaluator effect, especially in TA studies. The development of such criteria is, however, not easy as they are both system and task dependent and closely associated to the aim of the evaluation. Moreover, no matter how unambiguously the criteria are defined, applying them is, in the end, a matter of subjective judgement.

# 6 Conclusion

Based on a review of eleven studies of CW, HE, and TA, we have found that different evaluators evaluating the same system with one of these methods detect substantially different sets of usability problems in the system. This evaluator effect persists across differences in system domain, system complexity, prototype fidelity, evaluator experience, problem severity, and with respect to detection of usability problems as well as assessments of problem severity. In the reviewed studies, the average agreement among any two evaluators ranges from 5%-65%, and no one of the UEMs is consistently better than the others. The agreement among two evaluators is the relationship between the number of problems they have in common and the number of problems they have collectively detected. As a measure of the evaluator effect, we prefer the any-two agreement to the more widely reported detection rate because the detection rate is difficult to interpret correctly and measures coverage rather than agreement.

We believe that the principal cause for the evaluator effect is that usability evaluation is a cognitive activity, which requires that the evaluators exercise judgement. Thus, complete agreement among evaluators is unattainable. As we consider usability evaluation pertinent to the development of usable systems, we are however concerned about the magnitude of the evaluator effect in currently available UEMs. A substantial evaluator effect may not be surprising for a UEM as informal as HE but it is certainly notable that only marginally better agreement among the evaluators is achieved by adding the strict procedure of CW and by observing users who think out loud. Three aspects of the methods are considered as contributors to the evaluator effect: (1) vague goal analysis, (2) vague evaluation procedures, and (3) vague problem criteria. Several of the reviewed studies have dealt with one of the three vaguenesses and can serve to illustrate that as long as the other vaguenesses remain, the evaluator effect is still substantial. A couple of the studies attempt to deal with all three vaguenesses and achieve some of the most consistent results, though better agreement must still be a top priority.

## 6.1 Open research questions

*Do UEMs produce valid results?* The evaluator effect makes it apparent that evaluators disagree on what problems an interface contains but it does not tell whether this is due to real problems that are not reported (misses) or reported problems that are not real (false alarms). In most evaluations of UEMs, the issue of false alarms receives no consideration as any problem report is accepted as a usability problem. This leaves us virtually without evidence on which of the reported problems that matters to actual users doing real work (see also the discussion in Gray & Salzman, 1998; Olson & Moran, 1998). Specifically, we do not know whether evaluators should be advised to apply a higher threshold before they report a problem – to avoid false alarms – or a lower threshold – to avoid misses.

*Is the evaluator effect a result of inter-evaluator variability or intra-evaluator variability?* We need to investigate whether the evaluator effect reflects a true disagreement among evaluators or owes to inconsistencies in individual evaluators' performance. None of the reviewed studies have investigated whether the evaluators are consistent across evaluations. Hence, the evaluator effect as discussed in this study comprises inter-evaluator variability as well as intra-evaluator variability, and we do not know how much each contributes to the overall evaluator effect. The distinction between these two types of variability may be important because they may have different causes.

## 6.2 Consequences for practitioners

*Be explicit on goal analysis and task selection.* Even for moderately complex systems it is prohibitively demanding in time and resources to evaluate all aspects of a system in one test. Thus, before doing any usability evaluation we should thoroughly analyse the goals of the evaluation and carefully select task scenarios: What should this particular evaluation tell us? What aspects of the system should be covered? What should these parts of the system support the user in doing? Who will use the system, and in what contexts? After the task scenarios have been made their coverage should be checked and, if necessary, the scenarios should be iteratively

improved. This process is intended to both strengthen the preparation phase in order to increase the impact of the evaluation and to ensure agreement among the involved parties as to what the evaluation is to achieve.

*Involve an extra evaluator, at least in critical evaluations.* If it is important to the success of the evaluation to find most of the problems in a system, then we strongly recommend using more than one evaluator. For TA and possibly other user-involving UEMs, it seems that a reduction in the number of users can somewhat compensate for the cost of extra evaluators without degrading the quality of the evaluation. Further, the multiple evaluators can work in parallel and thus may save calendar time compared to a single evaluator because the single evaluator needs to run more users. Having just two evaluators will both improve the robustness of the evaluation and provide an opportunity for the evaluators to experience for themselves to what extent they disagree and on what types of issues.

*Reflect on your evaluation procedures and problem criteria.* The currently available UEMs are not as reliable as we would like them to be. Hence, much is left to personal judgement and work routines established among colleagues. This means that much can be learned from periodically taking a critical look at one's practices to adjust the evaluation procedure, tighten up problem criteria, and so forth. Peer reviewing how colleagues perform usability evaluations seems a valuable source of input for discussions of best practices and a way of gradually establishing a shared notion of usability and usability problems.

Finally, in spite of the evaluator effect, usability evaluations are a prerequisite for working systematically with ensuring and improving the usability of computer systems. Although the UEMs reviewed in this paper are not perfect, we still believe they are among the best techniques available.

## Acknowledgements

Morten Hertzum was supported by a grant from the Danish National Research Foundation. We wish to thank Iain Connell for providing us with additional data from Connell and Hammond (1999), Hilary Johnson for access to the data set from Dutt et al. (1994), and Rolf Molich for making the data set from Molich et al. (1999) available on the Web (http://www.dialogdesign.dk/cue.html). For insightful comments on earlier versions of this paper we wish to thank Iain Connell, James Lewis, Rolf Molich, and John Rieman.

## References

Connell, I.W., & Hammond, N.V. (1999). Comparing usability evaluation principles with heuristics: Problem instances vs. problem types. In M. Angela Sasse & C. Johnson (Eds.), *Proceedings of the IFIP INTERACT'99 Conference on Human-Computer Interaction* (pp. 621-629). Amsterdam: IOS Press.

Corona, R., Mele, A., Amini, M., De Rosa, G., Coppola, G., Piccardi, P., Fucci, M., Pasquini, P., & Faraggiana, T. (1996). Interobserver variability on the histopathologic diagnosis of cutaneous melanoma and other pigmented skin lesions. *Journal of Clinical Oncology*, 14(4), 1218-1223.

Cramer, S.F. (1997). Interobserver variability in dermatopathology. *Archives of Dermatology*, 133(8), 1033-1036.

Dumas, J.S., & Redish, J.C. (1993). *A practical guide to usability testing*. Norwood, NJ: Ablex.

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* (pp. 109-121). Cambridge: Cambridge University Press.

Funk, M.E., Reid, C.A., & McCoogan, L.S. (1983). Indexing consistency in MEDLINE. *Bulletin of the Medical Library Association*, 71(2), 176-183.

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.

Hertzum, M., & Jacobsen, N. E. (1999). The evaluator effect during first-time use of the cognitive walkthrough technique. In H.-J. Bullinger & J. Ziegler (Eds.), *Human-Computer Interaction: Ergonomics and User Interfaces. Proceedings of the HCI International '99* (Vol. I, pp. 1063-1067). London: Lawrence Erlbaum.

Jacobsen, N.E., Hertzum, M., & John, B.E. (1998). The evaluator effect in usability studies: Problem detection and severity judgments. In *Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting* (pp. 1336-1340). Santa Monica: HFES. [A reduced version of this paper appears as: Jacobsen,

N.E., Hertzum, M., & John, B.E. (1998). The evaluator effect in usability tests. In *Summary Proceedings of the ACM CHI 98 Conference* (pp. 255-256). New York: ACM Press]

Jacobsen, N.E., & John, B.E. (1999). A tale of two critics: Case studies in using cognitive walkthrough. Manuscript submitted for publication.

Lesaigle, E.M., & Biers, D.W. (2000). Effect of type of information on real time usability evaluation: Implications for remote usability testing. In *Proceedings of the XIVth Triennial Congress of the International Ergonomics Association and 44th Annual Meeting of the Human Factors and Ergonomics Society* (pp. 6-585 - 6-588). Santa Monica: HFES.

Lewis, C. (1982). *Using the 'thinking-aloud' method in cognitive interface design*. IBM Research Report RC 9265 (#40713). Yorktown Heights, NY: IBM Thomas J. Watson Research Center.

Lewis, C., Polson, P., Wharton, C., & Rieman, J. (1990). Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. In *Proceedings of the ACM CHI'90 Conference* (pp. 235-242). New York: ACM Press.

Lewis, C., & Wharton, C. (1997). Cognitive walkthroughs. In M. Helander, T.K. Landauer, & P. Prabhu (Eds.), *Handbook of Human-Computer Interaction. Second, completely revised edition* (pp. 717-732). Amsterdam: Elsevier.

Lewis, J.R. (1994). Sample sizes for usability studies: Additional considerations. *Human Factors*, 36(2), 368-378.

Lewis, J.R. (this issue). Evaluation of procedures for adjusting problem-discovery rates estimated from small samples.

Miller, J., Wood, M., & Roper, M. (1998). Further experiences with scenarios and checklists. *Empirical Software Engineering*, 3(1), 37-64.

Molich, R., Bevan, N., Curson, I., Butler, S., Kindlund, E., Miller, D., & Kirakowski, J. (1998). Comparative evaluation of usability tests. In *Proceedings of the Usability Professionals Association 1998 Conference* (pp. 189-200). Chicago, IL: UPA.

Molich, R., Thomsen, A.D., Karyukina, B., Schmidt, L., Ede, M., van Oel, W., & Arcuri, M. (1999). Comparative evaluation of usability tests. In *Extended Abstracts of ACM CHI 99 Conference*. New York: ACM Press.

Nielsen, J. (1992). Finding usability problems through heuristic evaluation. In *Proceedings of the ACM CHI'92 Conference* (pp. 373-380). New York: ACM Press.

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

Nielsen, J. (1994a). Heuristic evaluation. In J. Nielsen & R.L. Mack (Eds.), *Usability Inspection Methods* (pp. 25-62). New York: John Wiley.

Nielsen, J. (Ed.). (1994b). Usability laboratories [Special issue]. *Behaviour & Information Technology*, 13(1&2).

Nielsen, J., & Landauer, T.K. (1993). A mathematical model of the finding of usability problems. In *Proceedings of the INTERCHI'93 Conference* (pp. 206-213). New York: ACM Press.

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. In *Proceedings of the ACM CHI'90 Conference* (pp. 249-256). New York: ACM Press.

Olson, G.M., & Moran, T.P. (Eds.). (1998). Commentary on "Damaged merchandise?" *Human-Computer Interaction*, 13(3), 263-323.

Polson, P., & Lewis, C. (1990). Theory-based design for easily learned interfaces. *Human-Computer Interaction*, 5(2&3), 191-220.

Polson, P., Lewis, C., Rieman, J., & Wharton, C. (1992). Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies*, 36(5), 741-773.

Rowley, D.E., & Rhoades, D.G. (1992). The cognitive jogthrough: A fast-paced user interface evaluation procedure. In *Proceedings of the ACM CHI'92 Conference* (pp. 389-395). New York : ACM Press.

Sievert, M.C., & Andrews, M.J. (1991). Indexing Consistency in Information Science Abstracts. *Journal of the American Society for Information Science*, 42(1), 1-6.

Sørensen, J.B., Hirsch, F.R., Gazdar, A., & Olsen, J.E. (1993). Interobserver variability in histopathologic subtyping and grading of pulmonary adenocarcinoma. *Cancer*, 71(10), 2971-2976.

Virzi, R.A. (1992). Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors*, 34(4), 457-468.

Wharton, C., Bradford, J., Jeffries, R., & Franzke, M. (1992). Applying cognitive walkthroughs to more complex user interfaces: Experiences, issues, and recommendations. In *Proceedings of the ACM CHI'92 Conference* (pp. 381-388). New York: ACM Press.

Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994). The cognitive walkthrough method: A practitioner's guide. In J. Nielsen & R.L. Mack (Eds.), *Usability Inspection Methods* (pp. 105-140). New York: John Wiley.

Zunde, P., & Dexter, M.E. (1969). Indexing consistency and quality. *American Documentation*, 20(3), 259-267.

**Table 1**. Summary of results in the eleven reviewed studies. The detection rates for all problems and for severe problems only should not be compared across studies without having a closer look at the methodology used in each of the studies. A dash ('-') indicates that the figure could not be calculated from the available data.

| Reference | UEM | Evaluated system | Task scenarios | Total problems detected | Evaluators | Detection rate, all problems | Detection rate, severe problems | Any-two agreement |
|---|---|---|---|---|---|---|---|---|
| Lewis et al., 1990 | CW | Electronic mail | Yes | 20 | 4 (three of the developers of CW and a CE+ novice) | 65% | - | - |
| Dutt et al., 1994 [1] | CW | Personnel recruitment | Yes | 32 | 3 (two CS graduate students and a HCI researcher) | 73% | - | 65% |
| Hertzum & Jacobsen, 1999 | CW | Web-based library | Yes | 33 | 11 CS graduate students | 18% | 21% | 17% |
| Jacobsen & John, 1999 | CW | Multimedia authoring | No | 46 | 2 CS graduate students | 53% | - | 6% |
| Nielsen & Molich, 1990 [2] | HE | Savings | No | 48 | 34 CS students | 26% | 32% | 26% |
| | | Transport | No | 34 | 34 CS students | 20% | 32% | - |
| | | Teledata | No | 52 | 37 CS students | 51% | 49% | - |
| | | Mantel | No | 30 | 77 computer professionals | 38% | 44% | 45% |
| Nielsen, 1992 [3] | HE | Banking | No | 16 | 31 novices (CS students) | 22% | 29% | - |
| | | | | | 19 usability specialists | 41% | 46% | 33% |
| | | | | | 14 double specialists | 60% | 61% | - |
| Nielsen, 1994a [2] | HE | Integrating | Yes | 40 | 11 usability specialists | 29% | 46% | - |
| Connell & Hammond, 1999 [4] | HE | Hypermedia browser | No | 33 | 8 undergraduates[5] | 18% | 19% | 9% |
| | | | | 84 | 5 HCI researchers[5] | 24% | 22% | 5% |
| | | Interactive teaching | No | 57 | 8 psychology undergraduates[5] | 20% | 16% | 8% |
| Jacobsen et al., 1998 | TA | Multimedia authoring | Yes | 93 | 4 HCI researchers with TA experience | 52% | 72% | 42% |
| Molich et al., 1998 | TA | Electronic calendar | No | 141 | 3 commercial usability labs[6] | 37% | - | 6% |
| Molich et al., 1999 [7] | TA | Web-based email | No | 186 | 6 usability labs[8] | 22% | 43% | 7% |

Notes. [1] Hilary Johnson generously gave us access to the data set from Dutt et al. (1994).
[2] The detection rates for severe problems are reported in Nielsen (1992).
[3] The any-two agreement is calculated on the basis of data reported in Nielsen (1994a).
[4] Iain Connell generously gave us access to additional data from Connell & Hammond (1999).
[5] More evaluators participated in the study by Connell & Hammond (1999). We have extracted those using the ten HE heuristics.
[6] Four teams participated in the study by Molich et al. (1998) but only three of them used TA.
[7] Rolf Molich has generously made the data from the study available at www.dialogdesign.dk/cue.html
[8] Nine teams participated in the study by Molich et al. (1999) but only six of them used TA.

Footnotes

[1] In the earlier versions of CW the execution phase consisted of many more questions. However, the many questions made the walkthrough process inordinately tedious and time-consuming. In recognition of this, the execution phase of the latest version, described in Wharton et al. (1994) and Lewis & Wharton (1997), consists of only four questions.

[2] The heuristics have been slightly reworded to transform them from headings ("error prevention") to instructions ("prevent errors").

[3] The Kappa statistic is often used for measuring interrater agreement. However, Kappa presupposes – like the detection rate – that the total number of problems in the interface is known or can be reliably estimated. Since this is not the case for most of the studies reviewed in this paper (the number of evaluators is too small), we prefer to use the any-two agreement.

[4] Four teams participated in the study by Molich et al. (1998) but only three of them used TA.

[5] Nine teams participated in the study by Molich et al. (1999) but only six of them used TA.