What is Gained and Lost when using Evaluation Methods other than Empirical Testing

Heather W Desurvire, Jim M Kondziela & Michael E Atwood

NYNEX Science and Technology, Artificial Intelligence Laboratory, Intelligent Interfaces Group, 500 Westchester Avenue, White Plains, New York 10604, USA.
Tel: *+1 (914) 644 2757*

Fax: *+1 (914) 644 2211*

EMail: *hwd@nynexst.com*

There is increasing interest in finding usability testing methods that are easier and cheaper to implement than traditional laboratory usability testing. Recent research has looked at a few of these methods. The current study uses three groups of evaluators with different types of expertise, to evaluate a telephone-based interface using two different evaluation methods, the Cognitive Walkthrough and Heuristic Evaluation. This data is compared to laboratory results. Specific problems named in the laboratory and by the evaluator groups are analyzed for what contributions are made by each evaluator group under each method, and what is lost when traditional usability testing cannot be implemented. Future research directions are also discussed.

Keywords: Cognitive Walkthrough, Cost-effectiveness, Empirical Testing, Heuristic Evaluation, Telephone-based Interface, Usability Expertise, Usability Testing.

1. Introduction

There is increasing interest in finding usability testing methods that are easier and cheaper to implement than traditional laboratory usability testing, which is frequently not performed due to the lack of funds, planning, or human factors expertise. Recent studies are beginning to study and compare such techniques in the hopes that they can be utilized when empirical usability testing cannot be implemented. It is the hope of researchers in this area to learn

enough about these 'alternative' techniques' strengths and weaknesses to develop a toolkit that can be offered and utilized by industry. These methods include Heuristic Evaluation (Nielsen & Molich, 1990), the Cognitive Walkthrough (Lewis et al., 1990; Polson et al., 1992), Think Aloud methods such as Cooperative Evaluation (Wright & Monk, 1991a), and evaluation that utilizes Ergonomic Criteria (Bastien & Scapin, 1991; Scapin, 1990). For Heuristic Evaluation, Nielsen (1992) found that human-factors Experts were the best at finding an interface's usability problems, especially Experts who were also expert in the interface domain. Desurvire, Lawrence & Atwood (1991) found Experts' evaluations were the most reliable, and their best guess predictions were predictive of laboratory performance. Karat, Campbell & Fiegel (1992) similarly found that heuristic results were reliable and significantly predictive of laboratory data, yet empirical laboratory testing identified four and five times as many problems. Jeffries et al. (1991) found that via Heuristic Evaluation, more severe problems were found than with laboratory testing or via the Cognitive Walkthrough. This comparison study did, however, only utilized Experts in the heuristic condition, and Software Engineers in the Cognitive Walkthrough. We, thus, were interested to learn how predictive the methods were to laboratory results, but by using the same evaluator group types in both the heuristic and cognitive method conditions.

Jeffries et al. (1991) argue that even though their study found Heuristic Evaluation facilitated finding the more serious problems, it required the use of too many experts which is unarguably a rare commodity in most industrial settings. Wright & Monk (1991b) found that utilizing software engineers as evaluators was effective when utilizing the Cooperative Think Aloud method, due to their being able to directly observe the user's interface interactions. Software engineers as evaluators working on their own designs were more effective than those evaluating different designs, but were not effective at predicting user performance. Nielsen & Molich (1989) originally intended Heuristic Evaluation to be utilized by software engineers, as a cost savings measure. Jeffries et al. (1991) claimed that the Cognitive Walkthrough is a viable method for software engineers, because it leads them to the knowledge that users are assumed to have and the actual internal states of the system that are relevant to the user's interaction with it. Yet, in their study they were not, unfortunately, able to gain access to the original software engineers, and were also not able run the software engineer group through the heuristic method condition. Thus, we were interested in utilizing as our evaluators of both methods, software engineers, human-factors experts, and non-experts.

In addition, we wanted to further investigate the difference between group and individual evaluations. Karat, Campbell & Fiegel (1992) state that "...interaction-enhancing procedures may heighten group productivity" and "...groups do offer the possibility of more accurate judgments than individuals, especially when working on complex tasks". We were interested in looking at differences between individual and group evaluations for the heuristic method, to see if the group interaction process increased the number of predicted problems, and if any inaccurate predictions of problems would be deleted.

In summary, we wanted to set up a study where the methods and type of evaluator expertise would be compared to each other, and compare to empirical laboratory testing data. We were interested in learning not only how predictive each method by group was of laboratory performance, but also what each different category of information each type of evaluator, and each method brought to the evaluation both in terms of what we lost if we did not perform the laboratory tests, and also what we might gain.

2. Method

2.1. Design

The study compares Heuristic Evaluation and the Cognitive Walkthrough evaluation methods to laboratory testing results on a new telephone-based interface. Three evaluator groups were utilized in each method:

- Human-Factors Experts.
- Non-Experts.
- The original system's Software Engineers.

Each evaluator group had three members.

2.2. Laboratory

The study involved a traditional usability test of a telephone-based interface, utilizing a representative sample of 18 users, performing 6 tasks. The users ranged in age from 18 to 25, and approximately equally split by gender. All participants were employed in small businesses, and had experience with one to five telephone accessories such as speed dialing, and call waiting. The experimenter observing the trial noted problems and the corresponding problem severity coded by the Problem Severity Code and the perceived attitude of the user, coded by the Problem Attitude Scale. The experimenter also collected task completion data, error data, time to complete task, and the number of tries to complete the task.

2.2.1. Problem Severity Code (PSC)

This code is a 3-point scale. When a user encountered a problem, the experimenter coded it as follows:

1 = minor annoyance or confusion.
2 = problem caused error.
3 = caused task failure.

2.2.2. Problem Attitude Scale (PAS)

This scale is coded by the experimenter when a user encountered a problem. This scale is meant to reflect the user's attitude about a problem, as observed by the experimenter. This data is rarely collected, and is often only captured after the experiment in a self-report format. By that time, however, the user has often forgotten some of their frustration with the system. The scale is as follows:

1 = content with the system.
2 = frustrated with the system.
3 = wants to throw the system out the window.

2.3. Evaluators

Three different groups of evaluators were utilized in the Cognitive Walkthrough and the Heuristic Evaluation conditions. We utilized three individuals for every group, based on Nielsen & Molich (1990) and Nielsen's (1992) recommendations on the best group size. These groups were:

1. Human Factors Experts, who were identified as having advanced educational degrees, and more than 3 years experience in human–computer interaction.

2. Non-Experts, who had some experience using computer and telephone systems, such as voice mail and word processors, and were similar to those used as participants in the laboratory studies.
3. Software Engineers, who were the original designers of the system under evaluation.

Note that the same Software Engineers were utilized for both the heuristic and cognitive conditions, due to our only having three software engineers on the project. Each evaluator group in both methods was given approximately 3 hours to perform a complete evaluation.

2.4. Materials

The telephone-based interface being evaluated was conveyed to the evaluator groups by paper flow-charts and organized by task. Since in industry, Software Engineers often use flow-charts to design and communicate the system, we decided to utilize this format. Thus, we were avoiding the cost of developing a prototype, early on in the design cycle.

2.5. Alternative Usability Methods

The evaluators studied and learned the system via paper specifications. These specifications were flow charts that were modified to show each task flow separately. The evaluators were allowed to ask questions to clarify any questions they might have.

2.5.1. Cognitive Walkthrough

The Cognitive Walkthrough is an evaluation method that attempts to simulate the human–computer interaction, that is, the interaction between the user and an interface while the user is in the process of performing a task (Lewis et al., 1990; Polson et al., 1992). A series of questions are asked which attempt to facilitate the evaluator to see if the user's goals match the actions which are a result of the interface design. With this, potential problems are named, and predictions of the percentages of users who will have this problem are also generated.

For our evaluation of the Cognitive Walkthrough method, we utilized an automated version of this method developed on the HyperCard software, called the Automated Cognitive Walkthrough (ACW), developed by Rieman et al. (1991). The software was modified to address the specific interface we were studying. The ACW facilitated the organization of the cognitive walkthrough by generating all the appropriate questions and tasks, actions, and goals in the correct order, as well as generating a report of the results at the end.

The ACW facilitated our usage of this method, addressing a common barrier to utilizing the Cognitive Walkthrough which is that it is laborious to complete. Still, since we were only able to get a commitment for half a day from each of our evaluators, and it took approximately that to complete the ACW as we learned from piloting we were only able to collect data on three tasks, as opposed to six we collected in the heuristic method. Therefore, all comparative analysis between methods utilized these three tasks. The results of the additional three set of tasks from the heuristic evaluation not included in the analysis, were not significantly different from those included.

As recommended by the authors of the Cognitive Walkthrough, we utilized groups of evaluators, where three individuals made up one group. Each session was structured by an experimenter, but that was the extent of the experimenters involvement. Each member of

the group were firs
group proceeded th
upon by 2 out of
this, we will be ab
Evaluation and the

2.5.2. Heuristic E

Heuristic Evaluatio
set of heuristics by

The heuristic cond
individual part, the
heuristics they wer
those used in our fi
the usability princi

- Simple an
- Speak Use
- Minimize
- Be Consis
- Provide Fe
- Provide Cl
- Good Erro
- Prevent Er
- Provide Sh
- User Does

The heuristics wer
place. The evaluat
to critically study
used by the experi
problems they prec
PAS scales (see Sec
as violating a parti
another experiment

'Best Guess' predi
& Atwood, 1991)
was taken in the la
and error rates, tim
severity, and the us
error rate predictio
consensus.

The second part of
to learn if group dis
presenting their na

the group were first given the paper flow-charts of each of the tasks to study. Then the group proceeded through the ACW. The group named a problem as such if it was agreed upon by 2 out of 3 of the members of the team, and error rates were predicted. From this, we will be able to compare the named problems, and error rates with the Heuristic Evaluation and the laboratory results.

2.5.2. Heuristic Evaluation

Heuristic Evaluation is a method developed by Nielsen & Molich (1989) which utilizes a set of heuristics by which the evaluators must evaluate the interface.

The heuristic condition was set up into two parts, the individual, the group part. In the individual part, the members of the group were given a short lecture on the usability heuristics they were to use for evaluating the interface. The heuristics were the same as those used in our first study, see (Desurvire, Lawrence & Atwood, 1991), and are based on the usability principles from Smith & Mosier's (1986) work:

- Simple and Natural Language.
- Speak User's Language.
- Minimize Memory Load.
- Be Consistent.
- Provide Feedback.
- Provide Clearly Marked Exits.
- Good Error Messages.
- Prevent Errors.
- Provide Shortcuts.
- User Does Not Need Documentation.

The heuristics were clearly and boldly posted in the room where the evaluations took place. The evaluators were then given the set of 6 tasks via the flow charts, and asked to critically study the flow charts utilizing the heuristics. Using the same form as was used by the experimenter in usability tests (see next section), the evaluators named all problems they predicted users might make, then to rate each problem using the PSC and PAS scales (see Section 2.2 for a description). Later on, an experimenter rated each problem as violating a particular heuristic. These were later checked for inter-rater reliability by another experimenter.

'Best Guess' predictions, similar to those made in an earlier study (Desurvire, Lawrence & Atwood, 1991) were made; that is, evaluators were asked to predict the same data as was taken in the laboratory experiments. These included predictions of: task completion and error rates, time to complete the task, the number of tries to complete the task, error severity, and the user's perceived attitude due to a problem. For the Cognitive Walkthrough, error rate predictions were taken, where evaluators had to agree with a two out of three consensus.

The second part of the heuristic experiment allowed the group members to interact in order to learn if group discussion had any effect on the evaluators' answers. Evaluators took turns presenting their named problems, and other group members were encouraged to refute or

	Did Occur	Potential	Improvements
Lab Total Number	25	29	31
Heuristic Evaluation			
Experts	44% (11)	31% (9)	77% (24)
Software Engineers	16% (4)	24% (7)	3% (1)
Non-Experts	8% (2)	3% (1)	6% (2)
Cognitive Walkthrough			
Experts	28% (7)	31% (9)	16% (5)
Software Engineers	16% (4)	21% (6)	3% (1)
Non-Experts	8% (2)	7% (2)	6% (2)

Table 1: Percentage of Problems Evaluators Found that Did Occur in the Lab, Could Potentially Occur, and Suggested Improvements to the Interface

agree with the claims. The evaluators then recorded whether they would either delete or add a problem to their list, or change any of their predictions as a result of the interaction.

3. Results

3.1. Evaluation Methods vs. Laboratory

The following tables represent the percentage that each evaluator, by evaluator group, predicted the total number of problems that were found in the laboratory-based usability study. Also shown are those problems that were named by evaluators as potential problems in the laboratory, and suggested improvements to the interface. Those issues named as potential problems that were not possible to occur, and improvements that were not feasible were thrown out of the totals. For this reason and due to overlap, the totals will not add up to 100 percent.

3.1.1. Named Problems

The Experts heuristically evaluating the interface's tasks found the highest percentage of problems that actually occurred in the laboratory (44%), followed by the Experts using the Cognitive Walkthrough (28%). The next best at predicting the laboratory problems were the Software Engineers in both methods. The performance of Software Engineers and Non-Experts did not interact with evaluation methods, see Table 1 for details.

The evaluators also listed problems that we categorized as improvements to the interface. Improvement are based on an evaluators' assumption that it may avoid a user's potential problem or annoyance. For example, "prompt <x> is too condescending to the user". Of all the improvements named by all the groups, the Experts using Heuristic Evaluation named the highest percentage at 77%. The other groups named even less (3% to 16%), (see Table 1 for details).

Named problems could also be categorized as those that were not possible in the interface. The Non-Experts in the Heuristic Evaluation condition were the only group to erroneously name these types of problems. A majority, 55%, of the total number of problems they named in the heuristic condition were those that could not occur in the system. When analyzing these named problems, it seems that there was a misinterpretation of the system.

Lab (25)
Heuristic Evalu
Experts
Software Enginee
Non-Experts
Cognitive Walkt
Experts
Software Enginee
Non-Experts

Table 2: Per

Lab (25)
Heuristic Evalu
Experts
Software Enginee
Non-Experts
Cognitive Walkt
Experts
Software Enginee
Non-Experts

Table 3: Percer
Experi

Finally, the evaluators
but that did not actual
may have arisen in tas
Evaluation condition c
groups (31%), similar
Software Engineers ha
at 25% and found 21%

3.1.2. Severity Predic

Experts were best at pre
Evaluation method, wh
found 18%. Software E
were better than the oth
see Table 2).

	Problem Severity Code (PSC)		
	Minor Annoyance/Confusion	**Problem Caused Error**	**Problem Caused Task Failure**
Lab (25)	(5)	(3)	(17)
Heuristic Evaluation			
Experts	80% (4)	67% (2)	29% (5)
Software Engineers	40% (2)	0% (0)	12% (2)
Non-Experts	20% (1)	0% (0)	6% (1)
Cognitive Walkthrough			
Experts	40% (2)	67% (2)	18% (3)
Software Engineers	0% (2)	0% (0)	12% (2)
Non-Experts	20% (1)	0% (0)	6% (1)

Table 2: Percentage of Laboratory Problems Predicted by Three Levels of Severity

	Problem Attitude Scale (PAS)		
	Still Content With System	**Frustrated With System**	**Wants to Throw System Out Window**
Lab (25)	(19)	(4)	(2)
Heuristic Evaluation			
Experts	37% (7)	25% (1)	100% (2)
Software Engineers	21% (4)	0% (0)	0% (0)
Non-Experts	11% (2)	0% (0)	0% (0)
Cognitive Walkthrough			
Experts	26% (5)	25% (1)	50% (1)
Software Engineers	16% (3)	25% (1)	0% (0)
Non-Experts	16% (3)	25% (1)	0% (0)

Table 3: Percentage of Laboratory Problems Predicted by Three Levels of Experimenter Perceived Users' Attitude of Problem

Finally, the evaluators also named problems that were determined to be potential problems, but that did not actually occur in the laboratory. For example, these were problems that may have arisen in tasks that were not tested in the laboratory. The Experts in the Heuristic Evaluation condition contributed the most to the total number of problems named by all the groups (31%), similar to the Experts in the Cognitive Walkthrough condition (31%). The Software Engineers had the next highest amount from the Heuristic Evaluation condition, at 25% and found 21% via the cognitive condition.

3.1.2. Severity Predictions

Experts were best at predicting problems that caused task failure, especially in the Heuristic Evaluation method, where they named 29%. Experts using the Cognitive Walkthrough found 18%. Software Engineers were next best, with a 12% prediction rate. The Experts were better than the other evaluator groups, at predicting problems of all severity types (see Table 2).

Group	Task Completion Rate
Lab	92%
Experts	83%
Software Engineers	100%
Non-Experts	100%

Table 4: Task Completion Rates and Predictions of them

Method	Group	Error Rates
Lab		36%
Heuristic Evaluation	Experts	42%
	Software Engineers	31%
	Non-Experts	26%
Cognitive Walkthrough	Experts	94%
	Software Engineers	23%
	Non-Experts	69%

Table 5: Error Rates and Predictions of them

Experts were also better at predicting subjects' attitudes about problems (see Table 3). That is, they were able to predict the most problems where the users had the worst attitude and "wants to throw system out of the window". Software Engineers were somewhat better than Non-Experts in predicting attitude.

3.2. *'Best Guess' Predictions*

For the Heuristic Evaluation condition, 'best guess' predictions were made. Only some 'best guess' predictions were actually predictive. All groups significantly predicted task completion rates, where Experts underestimated, and Software Engineers and Non-Experts overestimated, slightly (see Table 4). The Software Engineers correctly predicted the average number of tries to complete the tasks, and the other groups overestimated by an average of 1 and 1.5, respectively. All groups were not predictive of the clocked times to complete the tasks.

For the Heuristic Evaluation method, 'best guess' error rates were collected. The Cognitive Walkthrough method collected a similar prediction, where error rate predictions were collected over several actions, that would make up one task. These error rates of actions were averaged to total one error rate per task. In the Heuristic Evaluation condition, all groups predictions were fairly predictive of error rates. For the Cognitive Walkthrough method, only Software Engineers were fairly predictive of error rates, see Table 5.

3.3. *User-interface Related Categories*

The problems named by the evaluators were again sorted, but this time by the aspect of the user-interface it represented. These categories for this telephone-based interface are problems that effect or are effected by the:

- *Keying*: For example "order of key presses was 3,1,2, should be 1,2,3".

	Category of Problem				
	Key	**Time**	**Task**	**System**	**Prompt**
Lab (53)	**5**	**4**	**35**	**3**	**6**
Heuristic Evaluation					
Experts	20%	75%	11%	33%	33%
Software Engineers	0%	25%	3%	33%	17%
Non-Experts	20%	0%	6%	33%	0%
Cognitive Walkthrough					
Experts	0%	50%	12%	0%	33%
Software Engineers	0%	25%	0%	33%	33%
Non-Experts	0%	25%	3%	0%	0%

Table 6: Percentage of Problems Predicted, by Category

- *Time*: For example " it took too long to hear the beep".
- *Task*: For example "if the user dials 2, they'll never hear the <x> option".
- *System*: For example "the user should be allowed to also input <x.>, as they may be inclined to, but the system won't allow it".
- *Prompts*: For example "The prompt should say 'thank you'".

3.3.1. Predictions of Laboratory by Categories

Experts predicted 75% of the time related issues in the Heuristic Evaluation condition, and 50% of them in the Cognitive Walkthrough condition. All groups were able to predict 33% of the system related problems, except for the Cognitive Walkthrough Experts and Non-Experts. The Software Engineers were best at predicting time and system related issues, which is not surprising given their job emphasis. Experts were best at predicting time, system, prompt, and less so, task related issues. Non-Experts were poor at predicting all the categories, except for time issues in the Cognitive Walkthrough (25%), and System issues in the Heuristic Evaluation (33%). See Table 6.

3.3.2. Categories of Problems, by Evaluator Group

When looking at the total named problems of each group, regardless of their prediction of laboratory problems, there were some trends of focus each of the groups. Of note, in Heuristic Evaluation, Experts named the majority of their total problems in the prompt category, 68%. Software Engineers focused their problem set in the prompt and system related issues (58% and 25% respectively). Non-Experts named the majority of their problems in the task category. Of note in the Cognitive Walkthrough method, Experts named the majority of their problem sets in the prompt and keying categories (36% and 27% respectively). Software Engineers in the task and prompt categories (38% and 33%), and the Non-Experts again focused on task related issues (50%).

3.4. Occurrence of the Heuristics

The highest percentage of heuristics that were violated in the laboratory over all 6 tasks were, 'be consistent' (25%), and 'provide feedback' (27%), and to a lesser degree, 'minimize memory load' (17%) and 'prevent errors' (11%). For the heuristic condition, the proportion of each groups' named problems on the same tasks was looked at in order

to study differences of focus between the groups. Experts named the highest percentage of their problems as violating 'provide feedback' (37%), while Software Engineers found mostly violations of 'prevent errors' (43%). The Non-Experts were found to focus on both heuristics focused on by the other two groups; the majority of the problems was split between violations of the heuristic 'provide feedback' (26%) and 'prevent errors' (26%).

We found the our lowest occurrence of heuristics by groups to be the same as Nielsen (1992) found[1], 'good error messages' and 'clearly marked exits', where rates were only 0% to 1% We additionally had a low occurrence rate of naming the 'be consistent' heuristic (2%), while in the laboratory, 25% of the problems violated this heuristic. Contrary to Nielsen's finding of a low occurrence rate of 'prevent errors', we found a high occurrence rate. These findings make it an important recommendation to those using this method to emphasize these heuristics to evaluators.

3.5. *Effects of the Group Interaction on the Productivity and Accuracy of Problems Named by the Evaluators*

The effect of group social interaction has been viewed by many as having an enhancing effect due to its influencing more accurate judgments and heightened productivity (Hackman & Morris, 1989; Karat, Campbell & Fiegel, 1992). This was tested in the Heuristic Evaluation group, where all 6 tasks were analyzed. Members of the group were asked to list the problems they named to the other members, who were encouraged to refute or agree with them. For testing accuracy of judgments, only the Non-Experts and Software Engineers were used since these were the only groups who named erroneous problems. The Non-Experts reduced these erroneous problems by 2% via group discussion, and the Software Engineers did not reduce their list of erroneous problems as a result of the group discussion. .On the other hand, there is some evidence that there was a facilitation of productivity, where Experts added 16% to their final total number of problems, and Non-Experts added 15%. The Software Engineers did not delete or add any problems as a result of the group interaction.

4. Discussion

Since usability testing in industry cannot always be performed, due to limited resources, its use must be timed for the most beneficial and efficacious period in the design cycle of a product. Because of this, researchers have evolved 'alternative' methods for studying the usability of a product. It is generally agreed that usability testing in both field and laboratory, is far and above the best method for acquiring data on usability; however, when resources are tight, quicker and 'dirtier' methods are helpful if they provide useful information. This study, and those like it, attempt to determine what information they provide is useful, in what context, what other types of information are we getting, and how much can we trust the information we are getting. With this in mind, we presented the comparison of two evaluation methods, using various evaluators as they compare to the benchmark of laboratory results.

This study indicates that Heuristic Evaluation is a better method than the Cognitive Walkthrough for predicting specific problems that actually occur in the laboratory, especially

[1] We utilized the same heuristics as Nielsen (1992; Nielsen & Molich, 1989; Nielsen & Molich, 1990) utilized, except for the additional heuristic 'user does not need documentation'.

for Experts. However, Experts using Heuristic Evaluation identified only 44% of the problems observed in the laboratory. Experts in the Heuristic Evaluation condition named almost twice as many problems that caused task failure or were of minor annoyance in the laboratory, than Experts in the cognitive condition. The Cognitive Walkthrough, however, facilitated more predictive error rate predictions for Software Engineers, than for Experts and Non-Experts within that method. 'Best guess' predictions of task completion, made in Heuristic Evaluation were equally predictive across all evaluator types. This is contrary to Desurvire, Lawrence & Atwood's (1991) results where only Experts were reliable and significantly predictive of both error and task completion rates. Interestingly, Experts were more conservative than the other groups with these predictions in Heuristic Evaluation, and more liberal in the Cognitive Walkthrough. This may due to the difference in calculating these predictions under each method.

Heuristic Evaluation seems to facilitate the identification of potential problems and improvements that go beyond the scope of the tasks, more so than the Cognitive Walkthrough. This may be due to the heuristic method 'reminding' Experts to analyze more dimensions of the interface than does the Cognitive Walkthrough.

When looking at categories of the user-interface in the Cognitive Walkthrough evaluation, Experts are good at predicting time, task and prompt related problems. Software Engineers are good at naming system, time, and prompt related problems. Non-Experts are only good at finding time related problems. In Heuristic Evaluation, Experts are good at time, prompt, task, and system related problems. Software Engineers are good at system, time, and prompt problems, and Non-Experts are best at system and keying problems. In fact, the Non-Experts in the cognitive condition were the only group that was good at naming keying problems.

Experts focused on problems that violate the heuristic, 'provide feedback', where they are more focused on the user's interaction with the system than the Software Engineers who were more focused on violations of 'preventing errors' due to their system focus. Interestingly, Non-Experts equally named as many problems as 'providing feedback' and 'preventing errors'.

Experts were the best at predicting laboratory problems that caused task failure, errors, and caused confusion in the users. The Experts were better in the heuristic condition than in the Cognitive Walkthrough, and there were no differences between methods for the Software Engineers and the Non-Experts. Experts were also best at predicting the user's attitude as a result of a problem in the laboratory. In fact, they predicted all categories well, especially those problems that caused the user to want to 'throw the system out the window'. Again, Experts' predictions were better in the heuristic condition. Software Engineers and Non-Experts, on the contrary, were better predictors of problems that caused frustrations with the system in the cognitive condition.

The effect of facilitating group discussion in the heuristic condition showed some interesting results. After evaluators individually evaluated the system, they were encouraged to discuss and argue their findings with the other group members. There was some evidence of producing more problems via the group interaction process. Software Engineers were not effected by the group interactions.

We have learned that the Cognitive Walkthrough and Heuristic Evaluations produce differing results, which is sometimes dependent on the type of evaluator. It is evident that each method had its strengths and weaknesses. It would be interesting to also learn if the process of performing either method teaches evaluation skills that could be applied later. For example, if proceeding through the Cognitive Walkthrough gives an evaluator such as a Non-Expert some knowledge of the human-computer interaction conceptual model, it might facilitate the Heuristic Evaluation. Also, does Heuristic Evaluation teach an evaluator group usability knowledge that will boost evaluation results on the Cognitive Walkthrough?

Very little usability research includes field research comparisons to either laboratory data, or alternative evaluation method data. Just as alternative evaluation techniques fall short of laboratory studies, we might find that laboratory studies fall short of field results, where 'real' usability can be measured. We argue that although laboratory research is important for creating a controlled environment for clean results, field data is important in usability research, for gaining a realistic environment, which is not as possible in the laboratory. Our next phase of research will include field research. It will be interesting to compare our results to field data, for determining whether different types of expertise or method are more predictive of field data than laboratory data. It will also be valuable to perform a cost/benefit analysis (Karat, 1990), and to study other alternative usability methods such as Co-Operative Evaluation (Wright, Monk & Carey, 1991) and utilizing Bastien & Scapin's (1991; Scapin, 1990) Ergonomic Criteria for evaluations.

This study has shown that evaluation methods can identify a number of interface problems, and these methods are particularly useful by Experts. While they cannot replace expert knowledge nor eliminate actual laboratory testing, they have the potential to significantly reduce the time and cost for evaluation and the severity of problems that occur in the prototyping stage.

While this study provides information on the relative value of alternative evaluation techniques, it also provides important information on their absolute value. At best, these methods provide only 44% of the problems seen in a laboratory based usability study. We believe this to be the most valuable finding. Alternative evaluation techniques identify some problems, but they fall short of empirical usability studies.

Acknowledgements

We would like to thank John Rieman for his Automative Cognitive Walkthrough software, and for his help and useful comments in our editing it. We would also like to thank Jakob Nielsen, Claire-Marie Karat, and Jean McKendree for useful discussions and comments on earlier drafts of this paper. Additionally we wish to thank all those who generously gave of their time to participate in our study.

References

C Bastien & D L Scapin (1991), "A Validation of Ergonomic Criteria for the Evaluation of User Interfaces", *ACM SIGCHI Bulletin* 23 (4), pp.54–55.

H Desurvire, D Lawrence & M E Atwood (1991), "Empiricism versus Judgment: Comparing User Interface Evaluation Methods on a New Telephone-based Interface", *ACM SIGCHI Bulletin* 23 (4), pp.58–59.

J R Hackman & C G Morris (1989), "Group Tasks, Group Interaction Process and Group Performance Effectiveness: A Review and Proposed Integration", in *Advances in Experimental Social Psychology*, L Berfkowitz [ed.] #8, Academic Press.

R J Jeffries, J R Miller, C Wharton & K M Uyeda (1991), "User Interface Evaluation in the Real World: A Comparison of Four Techniques", in *Proceedings of CHI'91: Human Factors in Computing Systems (Reaching through Technology)*, S P Robertson, G M Olson & J S Olson [eds.], ACM Press, pp.119–124, New Orleans, 28 April–2 May.

C Karat (1990), "Cost-benefit Analysis of Iterative Usability Testing", in *Human–Computer Interaction — INTERACT'90*, D Diaper, D Gilmore, G Cockton & B Shackel [eds.], Elsevier Science (North Holland).

C Karat, R Campbell & T Fiegel (1992), "Comparison of Empirical Testing and Walkthrough Methods in User Interface Evaluation", (In preparation).

C Lewis, P Polson, C Wharton & J Rieman (1990), "Testing a Walkthrough methodology for Theory-based design of Walk-up-and-use Interfaces", in *Proceedings of CHI'90: Human Factors in Computing Systems*, J C Chew & J Whiteside [eds.], ACM Press, pp.235–241.

J Nielsen (1992), "Finding Usability Problems Through Heuristic Evaluation", in *Proceedings of CHI'92: Human Factors in Computing Systems*, P Bauersfeld, J Bennett & G Lynch [eds.], ACM Press, pp.373–380, Monterey, CA, 3–7 May.

J Nielsen & R Molich (1989), "Teaching User Interface Design based on Usability Engineering", *ACM SIGCHI Bulletin* 21 (1), pp.45–48.

J Nielsen & R Molich (1990), "Heuristic Evaluations of User Interfaces", in *Proceedings of CHI'90: Human Factors in Computing Systems*, J C Chew & J Whiteside [eds.], ACM Press, pp.249–256.

P Polson, C Lewis, J Rieman & C Wharton (1992), "Cognitive Walkthroughs: A Method for Theory-based Evaluation of User Interfaces", (Submitted for publication).

J Rieman, S Davies, D C Hair, M Esemplare, P G Polson & C Lewis (1991), "An Automated Walkthrough", in *Proceedings of CHI'91: Human Factors in Computing Systems (Reaching through Technology)*, S P Robertson, G M Olson & J S Olson [eds.], ACM Press, pp.427–428, New Orleans, 28 April–2 May.

D L Scapin (1990), "Organizing Human Factors Knowledge for the Evaluation and Design of Interfaces", *International Journal of Human–Computer Interaction* 2 (3), pp.203–229.

S L Smith & J N Mosier (1986), "Guidelines for Designing User Interface Software", The MITRE-Corporation, Software Report MTR–10090.

P Wright, A F Monk & T Carey (1991), *Cooperative Evaluation — The York Manual*, Department of Psychology, University of York, (Version 0.4, 1989; version 1.0, 1991).

P C Wright & A F Monk (1991a), "The Use of Think-aloud Evaluation Methods in Design", *ACM SIGCHI Bulletin* 23 (1), pp.55–71.

P C Wright & A F Monk (1991b), "A Cost-effective Evaluation Method for Use by Designers", *International Journal of Man–Machine Studies* 35 (4), pp.891–912.

EVA
Use

Haral

Institute
Interact
Center f
PO Box
Tel: +4
Fax: +4
EMail:

Dependin requirem increasin will allo evaluatio a startin conforma overcome considers computer and com some det

Keywords

1. Introduction

The evaluation of u
the new European E
working conditions
(EC) published a dir
workers (EWG, 19