Overcoming the Developer Mindset Barrier towards Usability Evaluations

Anders Bruun & Jan Stage

Aalborg University, Department of Computer Science
bruun@cs.aau.dk, jans@cs.aau.dk

Abstract. Previous studies show that some software development practitioners without a usability background experience difficulties in understanding users and accepting that usability problems exist in their software. Also, they do not prioritize fixing the problems identified by specialists. This barrier is denoted "developer mindset" and this paper empirically explores whether the barrier can be overcome by training three software development practitioners to conduct usability evaluations. Findings show that practitioners succeeded in identifying 80.5 % of the usability problems, which shows a high level of user understanding. Findings also reveal that the practitioners prioritized fixing 61 % of the problems and we found that they successfully removed 64 % in a new interface design. We conclude that this approach may pose a viable solution to overcome the barrier of developer mindset.

1 Introduction

The level of commitment devoted to fix identified usability problems is important in practice as it is one of the main parts in determining whether a usability evaluation has been a success or not (Wixon 2003). Previous studies have adopted the concept of "downstream utility" to determine the extent to which results from usability evaluations impacts the usability of a system, cf. (Law 2006; Sawyer et al. 1996).

A recent survey conducted in 39 Danish software companies identified considerable barriers to introducing usability evaluations within their development processes (Bak et al. 2008). The most prominent of these barriers is denoted "developer mindset". This describes the tendency that some software development practitioners without a usability background are having difficulties in understanding users and accepting that there are problems in their software. It also covers reluctance to prioritize fixing usability problems (Bak et al. 2008). The barrier of developer mindset is also found in an Italian study equivalent to the Danish one (Ardito et al. 2011). This barrier compromises the commitment devoted to fix identified usability problems, which in turn leads to a low downstream utility.

A possible solution to overcome this barrier may be to involve software development practitioners in the execution of usability evaluations. A study conducted by Høegh and colleagues supports this by showing that developers' awareness of usability problems was increased after they observed sessions of user tests (Høegh et al. 2006). A reason for the increased awareness was that the observations of user tests provided firsthand experiences of how the system was applied in the real world by real users (Høegh et al. 2006). Involvement can also go beyond observation, e.g. by letting such software development practitioners actively participate in the evaluation process (Bruun and Stage 2011; Häkli 2005). Previous studies have shown promising results on letting such practitioners participate actively in usability activities. Metzker and Offergeld describe findings from a case study where software development practitioners together with usability specialists performed contextual task analysis. This motivated the practitioners to emphasize usability aspects (Metzker and Offergeld 2001). In (Häkli 2005) this is taken a step further by presenting a study where similar practitioners were trained to conduct user based evaluations without the presence of a usability specialist. A recent literature review shows that the majority of work with regard to training novices in usability engineering methods apply university students as the empirical basis (Bruun 2010). Additionally, there are few empirically based papers that focus on applying this approach in industrial practices (Bruun 2010).

In this paper we present an exploratory study on training software development practitioners from industry to conduct user based usability evaluations. The paper extends previous research by examining whether this approach could be a viable solution to overcome the barrier of developer mindset.

The paper is structured as follows. First a description of the experimental method is provided. This is followed by a presentation of our findings and a discussion of these in relation to related work. Finally we provide the conclusions and point out avenues of future work.

2 Method

This section describes the experimental procedure followed within this study. The overall idea was to train the software development practitioners (henceforth also mentioned as "practitioners") to conduct usability evaluations and then evaluate two versions of the same system. They evaluated the first version of the system after which they spent 3 months fixing the problems identified. The time span of 3 months was selected so that the practitioners had sufficient time to fix the problems. After 3 months a second evaluation was conducted to determine the effectiveness of the fixes.

2.1 Training Courses

2.1.1 Basic training course

The practitioners participating in our study had no practical experience in conducting usability evaluations (further details are provided the participants section below). For that reason we focused on teaching a traditional user based evaluation with video analysis as described in (Rubin and Chisnell 2008). This two-day (14 hours) training course was held by the authors of this paper as a combination of presentations and exercises. To conclude the basic training course the practitioners were asked to do a homework assignment of analyzing five video clips from a previous usability evaluation. The practitioners spent an average of 8.8 hours on this task. The resulting problem lists were reviewed and we provided feedback to the practitioners on how to improve their problem descriptions.

2.1.2 Follow-up training course

The traditional usability evaluation method taught during basic training necessitates traversing several hours of video data, which require a considerable amount of resources. For this reason we also chose to train the practitioners in applying IDA, as this method is not based on reviewing video data. IDA is conducted at the end of an evaluation and involves the following steps, cf. (Kjeldskov et al. 2004):

1. **Brainstorm:** The test monitor and data loggers participating in the evaluation identifies the usability problems they can remember while one of them notes problems on a whiteboard.
2. **Task review:** The test monitor and data loggers review all tasks to recall additional problems that occurred.
3. **Note review:** The data loggers review their notes to remember further problems.
4. **Severity rating:** The test monitor and data loggers discuss the severity of the problems and rate these as critical, serious or cosmetic, cf. (Molich 2000).

This one-day (7 hours) follow-up course in IDA was held by the authors two months after the basic training. A combination of presentations and exercises was also applied in this course.

2.2 Participants

2.2.1 Software development practitioners

Three software development practitioners from a small Danish software company (< 25 employees) participated in this study. This company had previously taken part in the survey presented in (Bak et al. 2008) which showed the existence of developer mindset. Two of the practitioners worked as project managers but also had responsibilities as software developers while one worked as a software developer exclusively. Two had no previous knowledge of usability evaluations while one had theoretical knowledge from an HCI course taken during his education. Thus, none of them had previous practical experience in conducting usability evaluations.

2.2.2 Test users

A total of seven test users participated in the two evaluations, all of which were recruited by the practitioners. Three test users participated in the first evaluation and four in the second. The test users were employed as administrative staff within different companies and all had experience in applying for wage subsidies, a process which is supported by the evaluated system (the system is further elaborated in the system section below). None of them had used the system before.

2.2.3 Usability specialists

Three usability specialists analyzed the video material obtained from both evaluations in order to evaluate the performance of the practitioners. Two of these were external usability specialists employed in industry and the third was an HCI researcher (one of the authors). The external usability specialists had not otherwise taken part in the experiment. None of the specialists had previous experience from the domain of wage subsidies.

2.3 Conduction of the Usability Evaluations

This section describes how the two usability evaluations were conducted by the three software development practitioners. As mentioned previously, these evaluations were conducted 3 months apart.

2.3.1 *Planning*

The three practitioners planned and conducted the two evaluations including finding the test users as well as defining the 3 tasks given to these. The same 3 tasks were given in both evaluations. They also distributed the roles between them, i.e. who would act as test monitor and who would be the data loggers noting down problems during the evaluations.

2.3.2 *System*

The system evaluated was a web application used by administrative staff within companies to register and apply for wage subsidies. Applications filled out by the administrative staff are then submitted to the local municipality, which then provides companies with a subsidy for the employees enrolled in such a settlement. The system consisted of two parts: 1) A stepwise wizard in which the data would be entered and 2) a pdf form shown as a confirmation at the end of the wizard, in which users could edit previously entered data. The system was developed by the small software company in which the practitioners were employed.

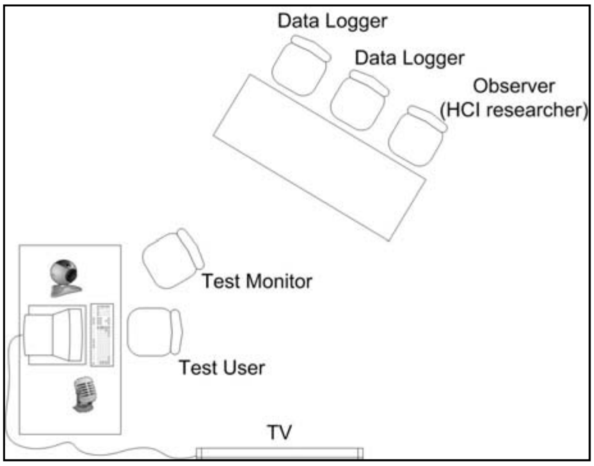

Figure 1: Overview of evaluation settings.

2.3.3 *Setting*

Figure 1 shows an overview of the setting applied in the evaluations, which were conducted in an office at the company. Video capture software was utilized to record user interaction within the system and a webcam recorded the face of the test users. An external microphone was applied to capture audio. During each session a test user was sitting at a table in the office solving the predefined tasks by using the system. One practitioner acted as test monitor and sat next to the test user. The two others acted as data loggers by noting down usability problems and observed the interaction through a projection on a 50” TV screen within the office. The data loggers along with one of the authors who observed the sessions sat 4 meters behind the test user in order not to interfere.

2.3.4 *Procedure*

Each of the two evaluations was conducted in one day. For each test session a user would firstly get introduced by the test monitor to the evaluation procedure and the system. Then the user was asked to solve the 3 tasks one while thinking aloud. If the user did not think aloud, the test monitor prompted her/him to do so. Each test session lasted 30 – 45 minutes.

At the end of each evaluation day the practitioners conducted Instant Data Analysis to derive the lists of usability problems. During both analysis sessions, one of the data loggers acted as facilitator by noting and organizing the identified usability problems on a whiteboard.

2.4 Improving the System

The list of usability problems from the first evaluation was used by the practitioners as input to improve the usability of the system. Two days after the first evaluation they held a one day meeting in which they discussed which problems to address and how to redesign the system. This was followed by three months of development, which was mainly done by the practitioner without project management responsibilities. All practitioners held weekly meetings during the period of development.

2.5 Interviews

After the second evaluation the three practitioners were interviewed by one of the authors. The interview was semi structured and was conducted with all practitioners present at once in order to foster discussions. The purpose was to uncover the factors influencing the practitioners' prioritization of fixing the problems and reasons for why some problems recurred in the second evaluation. Audio recordings from the interview was transcribed by one of the authors and analysed using grounded theory at open coding level (Strauss & Corbin, 1998).

2.6 Analysis of Problem Lists

The three usability specialists (one of the authors and two external) analyzed the seven video recordings from the two evaluations. Videos were analyzed in random order to reduce ordering bias. The specialists applied the same document template as the practitioners for describing problems. The severity of each problem was categorized as either cosmetic, serious or critical, corresponding to the categorizations applied by the practitioners. Analysis was firstly done individually where each specialist created two problem lists; one for the first version of the system and another for the second version. After the individual analysis all specialists held a meeting in which their lists were merged into two lists consisting of all identified problems (one total list for the first version of the system and a similar list for the second version). During the merging process the specialists negotiated the severity of the problems until agreement was reached. Finally the two lists created by the practitioners through the IDA sessions were merged into the lists created by the specialists. In case of identical problems, the severity ratings given by the practitioners were overridden by that given by the specialists. Severity ratings on problems identified by the practitioners only were not changed.

2.7 Measuring Developer Mindset

Developer mindset describes the tendency that some practitioners without a usability background are having difficulties in accepting results from usability evaluations as well as prioritizing these. This section describes the measurements utilized in our analysis to determine the level of developer mindset after training the practitioners. In our analysis we find the concept of “downstream utility” suitable for determining acceptance and prioritization. The downstream utility of an evaluation can be expressed in terms of the Committed Impact Ratio (CIR) and Completed-to-Date Impact Ratio (CDIR), cf. (Law 2006; Sawyer et al. 1996). CIR denotes the extent to which a development team commits to fixing usability problems before an actual implementation takes place and is calculated as follows:

$$CIR = \frac{\text{No. of problems comitted to fix}}{\text{Total no. of problems found}} \cdot 100$$

CDIR is a measure of the usability problems actually fixed at a given point, i.e. after implementation has begun:

$$CDIR = \frac{\text{No. of problems fixed}}{\text{Total no. of problems found}} \cdot 100$$

3 Results

This section presents our findings in relation to practitioner thoroughness in identifying usability problems as well as the downstream utility.

3.1 Thoroughness

Table 1 show an overview of the number of problems identified by the practitioners and specialists in the two evaluations. In total the practitioners and specialists identified 41 problems in the first version of the system (evaluation 1) and 44 in the second (evaluation 2). The practitioners identified 81 % and 80 % of all problems in evaluation 1 and 2 respectively while the specialists identified 76 % and 73 %.

	Evaluation 1		**Evaluation 2**	
	#	%	#	%
Practitioners	33	81	35	80
Specialists	31	76	32	73
Total	**41**	**100**	**44**	**100**

Table 1: Number of problems identified by practitioners and specialists in evaluation 1.

In both evaluations the practitioners identified fewer critical and more cosmetic problems than the specialists. In addition, findings reveal that the number of critical and serious problems was almost halved in the second version of the system while the number of cosmetic problems was doubled.

3.1.1 Differences in Severity Categorizations

The practitioners and specialists agreed on 23 of the 41 problems in the first evaluation, i.e. both groups identified the same 23 problems. The severity ratings (critical, serious or cosmetic) given by the two groups differed in 16 (70 %) of these problems where the practitioners consistently gave lower ratings than the specialists. In the second evaluation there was a disagreement on severity ratings in 5 of the 23 problems (22 %) found by both groups where the practitioners once more provided lower ratings than the specialists.

3.1.2 Uniquely Identified Problems

Across the two evaluations 22 problems were identified by the practitioners only. In the following we provide an example of one of the serious problems uniquely identified by this group. Two types of information are needed in the system in order to apply for wage subsidies. The first is related to the base salary, which includes the subsidy given by the municipality plus the amount paid by the employer while the other relate to the amount given by the employer only. During the evaluations the practitioners noted that some test users did only use the first type of information, which is not enough to submit a correct application form. The above mentioned example is highly domain specific and requires additional knowledge in order to be uncovered, especially since the users did not notice the problem themselves and, hence, did not comment on this explicitly during the evaluation. Other similar problems were identified by the practitioners but not by the usability specialists.

3.2 Committed Impact Ratio

The problem list made by the specialists was not available before starting the implementation, which meant that the practitioners only had access to their own list of problems derived through the IDA session from the first evaluation. For this reason we in the following base the total number of identified problems on the 33 found by the practitioners in evaluation 1. Before starting the implementation the practitioners committed to fix 20 problems. This gives the following CIR:

$$\text{CIR} = \frac{20}{33} \cdot 100 = 61\,\%$$

In the interview conducted at the end of the experiment we asked the practitioners of what factors had influenced the CIR, i.e. their prioritization of fixing usability problems. These factors were derived from existing literature and regarded: Severity ratings, frequency, length of problem descriptions and resource requirements, cf. (Law 2006).

The interviews revealed that the amount of resources required to fix a usability problem was one of the main factors in prioritizing these. One of the practitioners for instance mentioned that: *"... it didn't matter what severity rating the problems had but if it was a problem that could easily be corrected, it would come on the list of fixes"*. Another practitioner followed up on this by saying: *"Yes, and in the opposite case we have the problems which could cause great technical challenges. Those problems are on stand-by, not forgotten, but put into the log for future corrections"*.

Frequency in terms of the number of users experiencing a problem was not an influential factor, a practitioner for instance mentioned: *"If we have had ten test users and a problem was experienced by one of these it would be a different assessment compared to the three or four users we had... In our case we chose to say that if one user experiences the problem, it can also happen for others"*. Thus, problems experienced by three test users would not be emphasized over those experienced by a single test user.

The practitioners also mentioned that frequency, measured as the number of problems found within a given system component, influenced their priorities, e.g.: *"After our first* *evaluation we saw a lot of problems concerning the dates... The calendar component. It was that component that we spent the most time on improving"*.

The practitioners stated that the length of the problem descriptions did not influence their prioritization. As an example, one stated that: *"In the analysis we did not have problems where we said 'what was this problem?'"*, to which another replied: *"Yes, and the analysis was conducted immediately after the sessions so we did remember them"*.

An additional factor regarded the coherence to other systems in the company portfolio. The company was developing a new platform on which to base existing solutions, and if a usability problem was deeply rooted in the design of this old platform, it would not be prioritized, e.g.: *"... you can also say that what has happened in some cases was that we said 'but we will not fix this now as the new framework will come out later'"*. In Relation to this the practitioners only prioritized fixing problems related to the part of the system containing the stepwise wizard and not the editable pdf form.

In sum, the practitioners in our case mainly committed to fixing problems based on the factors of resource requirements and coherence to other systems while it did not matter whether a problem was experienced by a single or multiple test users. Finally, severity ratings and length of problem descriptions were less influential.

3.3 Completed-to-Date Impact Ratio

We found that 12 out of the total of 33 problems identified by the practitioners during the first evaluation recurred in the second version. Thus, 21 problems were fixed, which gives the following CDIR:

$$\text{CDIR} = \frac{21}{33} \cdot 100 = 64\,\%$$

During the interview we asked the practitioners why 12 problems from the first version of the system recurred in the second. One of the reasons was that 4 of these problems were related to the editable pdf form, which, as mentioned above, was not prioritized.

In case of the other 8 recurring problems, the practitioners mentioned they tried to fix 5 of these, but that the implemented fixes did not work as intended. One of the problems was related to the help texts, which lacked necessary information to which they mentioned: *“We have tried to make these more elaborate… We went through all of the texts to see if they properly explained the wordings”*.

The interviews also revealed that one of the recurring problems was not accepted by the practitioners as one of them mentioned: *“Well you could say that we should have taken this problem more seriously after the first evaluation, so we should have dug deeper into this already at the first test, just like we did after we found it again”*.

The final two recurring problems were not fixed as possible solutions conflicted with the usability of other system components or features prioritized by the sales department. As an example, one of these relates to the introduction presented in the system, which was not read by the test users due to its length. The solution of reducing the amount of text was not followed as this conflicted with the usability of another system component, one mentioned: *“With the introduction we also try to solve another problem about attachments. The introduction should avoid the users from getting stuck in the middle of the wizard because we let them know up front what attachments they need”*.

Summarizing on the above we found that the practitioners have tried to fix most of the prioritized problems that recurred, but that these fixes did not work as intended. Additionally, one of the problems was not accepted after occurring in the first evaluation, but was then prioritized after its presence in the second.

4 Discussion

Findings of our study show that the practitioners after receiving 30 hours of training were able to identify an average of 80.5 % of all usability problems across the two evaluations. In general, related work report of a lower thoroughness than the one found in our experiment. The study presented in (Bruun and Stage 2011) showed that 5 practitioners from industry on average were able to uncover 48.4 % of all usability problems and that two practitioners on average identified 71.4 %. In (Wright and Monk 1991) it is shown that each student team identified 33 % of all problems on average. In the study conducted by Koustabasis and colleagues it was found that students applying the user based method were able to identify 24 % of all problems on average (Koutsabasis et al. 2007). In (Frøkjær and Lárusdòttir 1999) it is shown that students were able to identify 18 % of all problems whereas the level of thoroughness reported in (Ardito et al. 2006) is lower as the students that applied a user based method identified an average of 11 %. The two studies presented in (Skov and Stage 2004; Skov and Stage 2008) compare student performance to that of specialists and show that students identified a mean of 37 % of the problems identified by specialists.

We see that the practitioners in our experiment were able to identify more problems than in the above mentioned studies. Differing motivational factors could play a considerable role in the variations between related work and our study. In a competitive market, software development practitioners are highly dependent on sales of their products, which is not the case for university students, which, with the exception of (Bruun and Stage 2011), are used in related work.

We also found that the three practitioners had a slighty higher thoroughness than the three specialists. A reason for this could be the thoroughness of the usability specialists within our study where each on average revealed 45 % of all problems. This, however, is comparable to the thoroughness presented in (Jacobsen et al. 1998) where four specialists conducting video based analysis identified an average of 52 % of all problems.

Another explanation of why the practitioners managed to identify more problems than the specialists could be that they had a higher level of domain knowledge. Our findings show that 22 problems were uniquely identified by the three practitioners and that some of those problems related to domain specific issues. This indicates one of the benefits of letting software development practitioners conduct evaluations compared to externally hired specialists, as their domain knowledge contributes in uncovering additional problems. This is upported in (Følstad and Hornbæk 2010) where a group of end users act as domain experts in the conduction of Coorporative Usability Evaluations. That study shows that evaluation output was enriched by including domain experts in the interpretation phase as they provided additional insight in identified problems and helped in uncovering a considerable amount of new problems (Følstad and Hornbæk 2010). The high performance of the software development practitioners within our study reveals that they have an elaborate understanding of the users of the system. This differs from the typical developer mindset presented in existing literature where it is reported that developers without usability knowledge have difficulties understanding their users, cf. (Ardito et al. 2011; Bak et al. 2008).

In terms of severity categorization we found a considerable disagreement between the ratings given by the practitioners and specialists. In the first evaluation there was a disagreement in categorization in 70 % of the problems found by both groups where the practitioners consistently gave lower severity ratings than the specialists. During the second evaluation there were disagreements in 22 % of the problems, giving an average of 46 % of all problems across both evaluations. This finding could indicate a downside to letting the practitioners evaluate their own systems, as they may be subjectively biased. This is supported within existing literature, e.g. in (Rubin and Chisnell 2008) where it is established that development teams for objectivity reasons should not evaluate their own designs. Although the practitioners' objectivity could be questioned in our case, we did find that they managed to uncover more problems than the specialists who had not taken part in the design or development of the system. A similar finding is presented in (Wright and Monk 1991) where it is shown that participants found more problems within their own designs than those made by others. Thus, the possible implications of objectivity may have a higher influence on severity ratings than on the number of problems identified.

Considering downstream utility we found that the practitioners before implementation primarily committed to fixing problems based on the factors of resource requirements and that severity ratings and frequency (no. of users experiencing a problem) did not matter. This prioritization resulted in a committed impact ratio of 61 %. Opposite findings are reported in (Law 2006), where it is found that severity ratings and frequency were the most influential factors on the impact ratio while the amount of resources required to fix the problems was less influential. This difference could be attributed to the fact that two of the practitoners in our study also had project management responsibilities, which could have caused an increased focus on resource requirements.

Considering downstream utility after implementing the second version of the system revealed no impact in terms of the number of identified usability problems. However, the number of critical and serious problems was reduced considerably as the count was almost halved, which indicates a highly positive impact on system usability. On the other hand, the number of cosmetic problems was almost doubled. The increase in cosmetic problems could be explained by the critical and serious problems, which may have masked the cosmetic ones during the first evaluation.

Although the number of identified problems remained on the same level, we found a completed-to-date impact ratio (CDIR) of 64 % as 21 of the 33 problems identified by the practitioners in the first evaluation was removed. CDIR is also revealed within other studies utilizing user based evaluations, in these, however, usability practices were already established as specialists were involved in evaluating and redesigning the systems. The study conducted by Medlock and colleagues revealed a higher CDIR of 97 % by applying the RITE method (Medlock et al. 2002). In (Hertzum 2006) the average CDIR is 65 %, which was obtained through the conduction of 5 user based evaluations. In (Law 2006) usability specialists conducted what is mentioned as "standard user tests" based on video analysis. In that study the CDIR was 38.3 %. Thus, the CDIR of 64 % found within our study resembles the one presented in (Hertzum 2006), which was obtained from a company with established usability practices and with employed usability specialists. On the other hand, this finding is lower than that reported in (Medlock et al. 2002). This could be explained by the fact that each team member in the Medlock study had limited responsibilities as e.g. usability engineer or developer. The practitioners in our case had more responsibilities besides conducting the usability evaluations, e.g. writing new code, fixing functionality problems and project management responsibilities.

Taken together, the above findings regarding downstream utility reveal that the software development practitioners within our study prioritized fixing the majority of usability problems as well as eliminating most of these in the following implementation. This indicates that their mindsets deviate from the typical developer mindset as reported in (Ardito et al. 2011; Bak et al. 2008) where developers did not accept or prioritize fixing problems. This deviation from the typical developer mindset is also supported by their high performance in identifying problems as mentioned previously as this indicates a high level of user understanding.

5 Conclusion

The purpose of this study was to examine whether training software development practitioners in conducting usability evaluations could be a viable solution to overcome the barrier of developer mindset. We found that the practitioners after receiving 30 hours of training were able to identify an average of 80.5 % of all problems across the two evaluations conducted, which indicates a high level of user understanding. Their performance was slightly higher than that of the usability specialists who uncovered an average of 74.5 %. A reason for this may be the practitioners' level of domain knowledge, which proves to be another advantage of letting them conduct evaluations. We also found that the practitioners committed to fixing 61 % of the identified problems and that they managed to eliminate 64 %, which resembles impact ratios found in other settings where usability practices already have been established. These impact ratios indicate that the practitioners accepted and prioritized most of the the problems, which deviates from the typical developer mindset found throughout existing literature.

The empirical data within our study is based on three practitioners from the same company conducting two evaluations, which makes it exploratory in nature. Further studies using more software development practitioners conducting evaluations are needed to validate our findings. Another limitation is that only one of the authors conducted an open coding analysis og the qualitative data where a more rigourous analysis would enrich and nuance findings of this study. Also, we have focused exclusively on training practitioners to conduct usability evaluations. As Wixon points out, then it is equally important to tell the practitioners what to do and not just what is wrong within an interface (Wixon 2003). Thus, in the future it would be crucial to provide such practitioners with training in interaction design to further increase the impact of usability evaluations.

6 Acknowledgements

The research behind this paper was partly financed by the Danish Research Councils (grant number 09-065143). We are grateful to the practitioners who participated and to the anonymous reviewers.

7 References

Ardito, C., Costabile, M.F., De Angeli, A., Lanzilotti, R. (2006). "Systematic evaluation of e-learning systems: an experimental validation", in *Proceedings of NordiCH*, ACM Press, New York.

Ardito C., Buono P., Caivano D., Costabile M.F., Lanzilotti R., Bruun A. and Stage J. (2011). "Usability Evaluation: A Survey of Software Development Organizations", in *Proceedings of SEKE*, Knowledge Systems Institute Graduate School, Skokie.

Bak, J. O., Nguyen, K., Risgaard, P. and Stage, J. (2008). "Obstacles to usability evaluation in practice: a survey of software development organizations", in *Proceedings of NordiCHI*, ACM Press, New York.

Bruun, A. (2010). "Training software developers in usability engineering: a literature review", in *Proceedings of NordiCHI*, ACM Press, New York.

Bruun, A. and Stage, J. (2010). "Evaluating the Usability of Home Healthcare Applications", in *Human Centered Design of E-Health Technologies*, Carsten Röcker and Martina Ziefle (Eds.), IGI Global, Hershey.

Bruun, A. and Stage, J. (2011). "Training Software Development Practitioners in Usability Evaluations: An Exploratory Study of Cross Pollination", in *Proceedings of PUX*.

Frøkjær, E., Lárusdòttir, M.K. (1999). "Prediction of Usability: Comparing Method Combinations", in *Proceedings of IRMA*, Idea Group Publishing, Hershey.

Følstad, A., Hornbæk, K. (2010). "Work-domain knowledge in usability evaluation: Experiences with Cooperative Usability Testing", *Journal of Systems and Software*, (83:11), pp. 2019-2030, Elsevier Science Inc. New York.

Gulliksen, J., Boivie, I., Persson, J., Hektor, A. and Herulf, L. (2004). "Making a difference: a survey of the usability profession in Sweden", in *Proceedings of NordiCHI*, ACM Press.

Häkli, A. (2005). *Introducing user-centered design in a small-size software development organization*, Helsinki University of Technology, Helsinki, 2005.

Hertzum, M. (2006). "Problem Prioritization in Usability Evaluation: From Severity Assessments Toward Impact on Design", *International Journal of Human-Computer Interaction*, (21:2), pp. 125-146, Taylor & Francis.

Howarth, J. (2007). *Supporting novice usability practitioners with usability engineering tools*, Virginia Polytechnic Institute & State University Blacksburg, Blacksburg, 2007.

Howarth, J., Andre, T.S., Hartson, R. (2007). "A Structured Process for Transforming Usability Data into Usability Information", *Journal of Usability Studies*, (3:1), pp. 7-23, UPA.

Høegh, R. T., Nielsen, C. M., Overgaard, M., Pedersen, M. B., Stage, J. (2006). "The Impact of Usability Reports and User Test Observations on Developers' Understanding of Usability Data: An Exploratory Study", *International Journal of Human-Computer Interaction*, (21:2), pp. 173-196. Taylor & Francis.

Jacobsen, N.E., Hertzum, M., John, B.E. (2007). "The Evaluator Effect in Usability Studies: Problem Detection and Severity Judgments", in *Proceedings of HFES*, HFES, Santa Monica.

Kjeldskov, J., Skov, M. B. and Stage, J. (2004). "Instant data analysis: conducting usability evaluations in a day", in *Proceedings of NordiCHI*, ACM Press, New York.

Koutsabasis, P., Spyrou, T., Darzentas, J.S., Darzentas J. (2007). "On the Performance of Novice Evaluators in Usability Evaluations", in *Proceedings of PCI*, Patra.

Law, E. (2006). “Evaluating the downstream utility of user tests and examining the developer effect: A case study”, *International Journal of Human-Computer Interaction*, (21:2), pp. 147-172, Taylor & Francis, London.

Medlock, M.C., Wixon, D., Terrano, M., Romero, R., Fulton, B. (2002). “Using the RITE method to improve products: a definition and a case study”, in *Proceedings of the Usability Professionals Association*, UPA.

Metzker, E., Offergeld, M. (2001). “An Interdisciplinary Approach for Successfully Integrating Human-Centered Design Methods Into Development Processes Practiced by Industrial Software Development Organizations”, in *Proceedings of the International Conference on Engineering for Human-Computer Interaction*, Springer-Verlag, London.

Molich, R. (2000). *User-Friendly Web Design*, Ingeniøren Books, Copenhagen, 2000.

Rubin, J., Chisnell, D. (2008). *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*, 2nd. Edition. John Wiley & Sons, Inc., Indianapolis, 2008.

Sawyer, P., Flanders, A. and Wixon, D. (1996). “Making A Difference – The Impact of Inspections”, in *Proceedings of CHI*, ACM Press, New York.

Skov, M. B. and Stage, J. (2004). “Integrating Usability Design and Evaluation: Training Novice Evaluators in Usability Testing”, in *Proceedings of the Workshop on Improving the Interplay between Usability Evaluation and User Interface Design*. NordiCHI.

Skov, M. B. and Stage, J. (2008). “Direct Integration: Training Software Developers and Designers to Conduct Usability Evaluations”, in *Proceedings of the First Workshop on the Interplay between Usability Evaluation and Software Development*, CEUR-WS.org.

Strauss, A., & Corbin, J. *Basics of qualitative research. Techniques and procedures for developing grounded theory (2nd ed.)*, Thousand Oaks, CA: Sage (1998).

Wixon, D. (2003). “Evaluating usability methods: why the current literature fails the practitioner”, in *Interactions*, (10:4), pp. 28-34, ACM Press, New York.

Wright, P.C., Monk, A.F. (1991). “The Use of Think-Aloud Evaluation Methods in Design”, in *ACM SIGCHI Bulletin archive*, (23:1), pp. 55 – 57, ACM Press, New York.