Taylor & Francis
Taylor & Francis Group

Behaviour & Information Technology

ISSN: 0144-929X (Print) 1362-3001 (Online) Journal homepage: http://www.tandfonline.com/loi/tbit20

Barefoot usability evaluations

Anders Bruun & Jan Stage

To cite this article: Anders Bruun & Jan Stage (2014) Barefoot usability evaluations, Behaviour & Information Technology, 33:11, 1148-1167, DOI: 10.1080/0144929X.2014.883552

To link to this article: http://dx.doi.org/10.1080/0144929X.2014.883552

Accepted author version posted online: 13 Jan 2014.
Published online: 17 Feb 2014.

Behaviour & Information Technology, 2014
Vol. 33, No. 11, 1148–1167, http://dx.doi.org/10.1080/0144929X.2014.883552

Taylor & Francis
Taylor & Francis Group

Barefoot usability evaluations

Anders Bruun* and Jan Stage

Department of Computer Science, Aalborg University, Selma Lagerlöfs Vej 300, DK-9220 Aalborg, Denmark

(*Received 29 June 2013; accepted 30 December 2013*)

Usability evaluations provide software development teams with insights on the degree to which a software application enables a user to achieve his/her goals, how fast these goals can be achieved, how easy it is to learn and how satisfactory it is in use. Although usability evaluations are crucial in the process of developing software systems with a high level of usability, their use is still limited in the context of small software development companies. Several approaches have been proposed to support software development practitioners (SWPs) in conducting usability evaluations and this paper presents two in-depth empirical studies of supporting SWPs by training them to become barefoot usability evaluators. Findings show that the SWPs after 30 hours of training obtained considerable abilities in identifying usability problems and that this approach revealed a high level of downstream utility. Results also show that the SWPs created relaxed conditions for the test users when acting as test monitors but experienced problems with making users think aloud. Considering the quality of problem descriptions, we found that the SWPs were better at providing clear and precise problem descriptions than at describing the impact, cause, user actions and providing data support for observations.

Keywords: training; developers; usability; evaluation

1. Introduction

Usability evaluations provide software development teams with insights on the degree to which a software application enables a user to achieve his/her goals, how fast these goals can be achieved, how easy it is to learn and how satisfactory it is in use (Rubin and Chisnell 2008). Although usability evaluations are crucial in the process of developing software systems with a high level of usability, their use is still limited in the context of small software development companies (Ardito *et al.* 2011).

Evaluating the usability of software applications can be accomplished through the use of several methods that can be categorised according to their empirical basis. Rubin and Chisnell (2008), for instance, emphasises user-based evaluations in which users are observed by usability specialists while they use an application to solve a set of pre-defined tasks and think aloud. Other evaluation methods are based on usability specialists or domain experts inspecting an interface in order to uncover potential usability problems, e.g. Heuristic Evaluation as proposed by Nielsen (1992).

There are several approaches to organise the responsibilities of conducting usability evaluations in the context of software development projects. One way is to apply an integrated approach where usability specialists, that are part of the software development team, act as evaluators of their own software (Høegh *et al.* 2006). Another is the separate unit approach in which usability specialists from a different organisational unit within the company conduct usability evaluations as a service to the development team (Høegh *et al.* 2006). Outsourcing denotes the third approach where usability specialists from another company are hired as external consultants to conduct usability evaluations (Høegh *et al.* 2006). The most common way to provide feedback from these approaches is a written report presenting the usability problems experienced by users (Høegh *et al.* 2006).

At least three approaches have been proposed to support SWPs in conducting usability evaluations: The first form of support is to provide SWPs with either software tools or conceptual tools to assist in identifying usability problems (Howarth 2007). The second approach is to provide support to SWPs through usability evaluation methods (Koutsabasis *et al.* 2007). The third approach is to support SWPs through training. In support of the third approach, Høegh and colleagues conducted a study of usability evaluation feedback formats in which they examine how to increase such practitioners' awareness of usability problems (Høegh *et al.* 2006). One of these feedback formats was to let the practitioners observe user-based evaluations to further involve them in the process (Høegh *et al.* 2006).

There are several causes for the limited application of usability evaluations in small companies. Perceived resource demands and developer mindset are two of the primary barriers for conducting usability evaluations

*Corresponding author. Email: bruun@cs.aau.dk

(Bak *et al.* 2008, Ardito *et al.* 2011). Perceived resource demands are a barrier especially present within small software companies as these do not have the funds to pay for comprehensive consultancy or staffing of usability specialists (Scholtz *et al.* 1998, Häkli 2005, Juristo *et al.* 2007), as they are expensive to hire (Nielsen 1994). The barrier of developer mindset reflects the situation that some SWPs (e.g. developers) experience difficulties in thinking like a user and what they are capable of (Bak *et al.* 2008). Developer mindset also covers the aspect of acceptance where problems identified through usability evaluations are not always accepted by the people in the organisation that did not participate in the conduction of the evaluations (Bak *et al.* 2008). Prioritisation of fixing identified usability problems is also part of the developer mindset where the implementation of functionality and fixing bugs receive higher priority (Bak *et al.* 2008). It can be argued that, if SWPs are able to conduct evaluations it could lessen the need for small companies to employ human–computer interaction (HCI) specialists. This could potentially solve issues in relation to funding. Also, letting SWPs conduct usability evaluations would provide them with first-hand observations of users, which in turn could be a solution to overcome the barrier of developer mindset.

1.1. Barefoot usability evaluators

As suggested above, one possible solution to support SWPs could be to train them to conduct evaluations and analyse the data. This is similar to the idea behind the barefoot doctors that emerged during the Cultural Revolution in China in the 1960s (cf. Daqing and Unschuld 2008, Bruun 2011). Getting health-care services embedded in the rural areas of China was an ongoing challenge dating back to the early twentieth century (Daqing and Unschuld 2008). Early attempts of solving this challenge included drafting doctors from private practices, but health-care services in these areas remained scarce. Mao Zedong criticised this urban bias of health-care services, and in 1965 he emphasised the importance of solving this challenge. To counter this problem, Mao sent mobile teams of doctors into these areas with the purpose of training local peasants in basic medicine, such as the delivery of babies, how to ensure better sanitation and how to perform simple surgical procedures (Daqing and Unschuld 2008). In order to keep up the level of mass production, peasants, who received medical training, would generate work points from their medical services as well as they would receive points for doing agricultural work. Thus, some of the peasants would work part-time in the rice fields walking around barefooted and part-time as doctors in the local area, which coined the term of barefoot doctors.

1.2. Objectives

In this paper, we present two studies of training SWPs from industry, that had no or minimum previous experience in usability work, to conduct usability evaluations. The aim is to evaluate the performance of the barefoot usability evaluators and compare this to the performance of HCI specialists. More specifically, our objective is to measure and discuss the performance of internal metrics in relation to the conduction of evaluation methods and following analysis as well as an external metric in terms of actual impact on an evaluated system. This leads to the following four research questions:

> RQ1. To what extent are software practitioners with minimal training in usability evaluations able to identify usability problems compared to the performance of HCI specialists?
>
> RQ2. How do the problems identified by software practitioners differ from those found by HCI specialists?
>
> RQ3. How are usability evaluations by software practitioners conducted, compared to best-practice? In particular, how do software practitioners perform as test monitors?
>
> RQ4. What is the level of 'downstream utility' of usability evaluations conducted by software practitioners?

We start by providing an overview of related work followed by descriptions of the empirical method. We then present and discuss our findings with respect to test monitor performance, thoroughness, any-two agreement, quality of problem descriptions, downstream utility and cost effectiveness. Finally, we present our conclusions and limitations of the study.

2. Related work

To uncover related work, we made a comprehensive literature survey of research conducted in the area of training novices in usability engineering (UE) methods. Papers were selected as relevant if they described or focused on training of novices in UE methods. We define 'novices' according to Bonnardel *et al.* (2003) and Howarth *et al.* (2007):

> Novices are persons with less than one year of job experience related to UE and no formal training in Usability Engineering methods

A preliminary screening was performed using Google Scholar, as this search engine covers scientific papers from a broad set of publishers and proceedings. The search criteria were based on a full-text search in which all the words 'training', 'developers' and 'usability' were required and resulted in 33,800 records (search was conducted on 1 December 2009). As it would be too tedious a task to read abstracts from all these papers, the first 200 abstracts were read of which 23 potentially relevant papers were selected and read in full. Eight papers were selected as relevant and defined the result of the screening process. Papers referenced in the selected eight papers from the screening were marked as potentially relevant. In addition, the eight relevant papers were looked up on Google Scholar, which provides a utility to identify which papers are citing these. All citations were

also marked as potentially relevant. Subsequently abstracts from all referenced and cited papers were read and papers fitting the selection criteria were read in full. This process continued until closure was reached after eight iterations. A total of 4155 abstracts were read and 286 papers were read in full ending up with 129 actually relevant papers, see Bruun (2010) for further details on this study. Papers were analysed in terms of research focus, empirical basis, types of training participants and training costs.

Half of the identified papers focuses on the development and evaluation of university curricula while the other half emphasises training in UE in relation to industry practice. Most of the practice-oriented papers focus on training novices in isolation from the organisational context, e.g. training of university students rather than SWPs from industry. Of the 129 papers we found, 13 consider the organisational context of the training, such as company size, staffing and development processes, and 3 of these are empirical studies.

The term UE applied above covers analysis, design and evaluation methods and we found that the majority of papers emphasise training in the latter, which is also the case for this paper. We recognise that training in methods related to analysis and design activities is also important but the literature suggests that results from usability evaluation methods are effective in creating the wake-up calls necessary for companies to start focusing on UE and to increase the awareness of developers (Høegh *et al.* 2006, Schaffer 2007). Fonseca *et al.* (2009) and Edwards *et al.* (2006) also describe university curricula where courses start by letting students evaluate user interfaces, as this increases their awareness of interface problems. The survey described in Rosenbaum *et al.* (2000) also shows that usability evaluations within and without a lab are the most preferred methods by software companies.

A closer look at papers emphasising evaluation methods reveals that 33 papers present training in non-user-based evaluation methods, such as heuristic inspection and cognitive walkthrough, while 11 papers represent training in user-based usability evaluations. Several studies have indicated that user-based evaluations outperform inspection and walkthrough methods with respect to awareness. Høegh *et al.* (2006), for instance, argue that user-based evaluations provide valuable first-hand observations of the problems experienced by users, which in turn increases the motivation for making adjustments to the user interface. This may be explained by the fact that user testing provides empirical evidence of the problems at hand compared with theoretical inspections (Brown and Pastel 2009). Additionally, the study conducted by Frøkjær and Hornbæk (2008) indicates that a majority of evaluators prefer user-based methods over the inspection methods, Metaphors of Human-Thinking and Cognitive Walkthrough. Finally, in the study presented in Ardito *et al.* (2006) it was found that user-based methods were rated above inspection methods with regard to pleasantness of use. Thus, the above studies informed us to emphasise user-based usability evaluations with respect to training of novices.

2.1. *User-based evaluation studies*

This section provides an overview of 11 empirical studies where novice usability evaluators conducted user-based usability evaluations. We have identified three research foci in these papers: Studies of tools, studies of methods and studies of training.

2.1.1. *Studies of tools*

Three papers present studies of either software tools or conceptual tools that assist evaluators in identifying usability problems. Two papers are based on the same experiment and describe the development and evaluation of a software tool aiming to ease transformation of raw usability data into usability problem descriptions (Howarth 2007, Howarth *et al.* 2007). Sixteen graduate students participated as usability evaluators and applied either of two software tools to note problems. Participants received one-hour training to get acquainted with the tools after which they were asked to view videos from a previously conducted usability evaluation. Participant performance was measured in terms of the quality of the problem descriptions they provided. Quality criteria are based on the work by Capra (2006) and regard clarity of problem descriptions, severity, data support, problem cause and user actions. Results show that students were better at formulating user actions than providing clarity, data support, etc. in their problem descriptions.

Skov and Stage (2005) present a study on developing and evaluating a conceptual tool to support problem identification. This tool is represented by a 4×3 matrix to be applied by evaluators when observing users. A comparative study based on 14 undergraduate students was conducted and participants were distributed over two experimental conditions; one in which the conceptual tool was applied to test a user interface and another without the tool. Students were then asked to view recordings from a previous evaluation of a user applying a web-based system to solve a series of tasks. Findings in that study related to the number of problem identified and showed that students were able to identify 18% of all problems and that they discovered a mean of 20% of the problems identified by two usability specialists.

2.1.2. *Studies of methods*

Three papers present comparative studies of usability evaluation methods. Koutsabasis *et al.* (2007) describe an experiment in which the focus is on evaluating the performance of students in terms of number of identified problems, validity and efficiency. The empirical basis of that study is 27 students, which were distributed over the four conditions of Heuristic Inspection, Cognitive Walkthrough, user-based

evaluation and Co-discovery learning. Results show that students applying the user-based method were able to identify 24% of all problems on average (Koutsabasis *et al.* 2007).

Ardito *et al.* (2006) describe the development and evaluation of the e-learning systematic evaluation (eLSE) method for evaluating e-learning systems. In that study, 73 senior students were used as the empirical basis for comparing the performance of the following methods: eLSE, user-based evaluation and Heuristic Inspection. Results here related to the number of identified problems and findings show that students applying the user-based method identified an average of 11% of all problems.

In Frøkjær and Lárusdòttir (1999), a comparative study of usability evaluation methods is presented. That paper emphasises the effect of combining methods and 51 students participated in the study. Participants were in the first condition asked to apply Cognitive Walkthrough followed by the second round applying a user-based evaluation method. In the second condition, other participants applied Heuristic Inspection followed by user-based evaluation. Results from that study show that students were able to identify 18% of all problems.

2.1.3. *Studies of training*

Five papers emphasise training of novices in analysing data from user-based evaluations. Three of these papers are based on the same experiment in which 36 teams of first-year students were trained in how to conduct usability tests (Skov and Stage 2004, Skov and Stage 2008, Skov and Stage 2009). Students received 40 hours of training before participating in the experiment and they were instructed to conduct an evaluation, analyse the results and to write a report documenting all steps in the process. The reports written by the students were compared to that written by eight teams of usability specialists. Findings show that students uncovered a mean of 7.9 problems and that specialists found a mean of 21, i.e. the students on average found 37% of the problems identified by specialists.

Wright and Monk (1991) describe two experiments studying the application of user-based evaluations. The first experiment concerns the effectiveness of usability evaluation when applied by software trainees after reading a short manual. Trainees documented identified problems in a report that was assessed in terms of the number of identified problems and severity. Trainee performance was then compared to that of usability specialists. The second experiment examines differences of evaluating own design versus the design made by others and two new groups of trainees were divided into two conditions. In the first condition, trainees designed and evaluated their own prototype and in the second they evaluated designs made by others. Again, reports were assessed with respect to the number of identified problems. Results indicate that all student teams identified 33% of all problems on average.

The final related work paper is a master thesis describing efforts aiming to introduce a user-centred method in a small software company (Häkli 2005). The second purpose was to increase the knowledge of software developers on the matter. In that study, a 14-hour training course for 13 SWPs was conducted. The contents of the course were related to the topics of interaction design, prototyping, Heuristic Inspection and user-based evaluation. That study primarily focused on the participant performance in conducting Heuristic Inspections. However, some qualitative observations on how well the participants performed as test monitors in a user-based evaluation were collected. It is mentioned that the test conduction went 'quite nicely' although it was 'rather unmanaged' (Häkli 2005). It was also observed that the participants acting as test monitors were unable to keep the test on track and that they rarely encouraged users to think aloud, which in turn led to much of the test being conducted in silence.

2.1.4. *Research needs*

Summarising on the related work, it can be seen that the main focus of these papers is evaluating the performance of university students in conducting usability evaluations. A notable exception to this is Häkli (2005) that uses SWPs from industry as the empirical basis. This leaves room for further studies on the ability of software practitioners to identify usability problems, a need which is also supported in Skov and Stage (2009). In relation to this, Wixon argues that results of usability evaluation studies may not be of practical significance if assessed in isolation from organisational contexts (Wixon 2003).

Another point of consideration is the fact that the majority of papers focus on quantitative aspects of usability evaluation, such as the number of problems identified. Thus, there is a need to further report findings on aspects, such as the quality of problem descriptions, and in particular which parts of the descriptions that novices find difficult, e.g. clarity, impact, data support, cause and user actions (cf. Capra 2006). Furthermore, Wixon (2003) mentions that 'problem should be fixed and not just found'. The point made here is that it is relevant to go beyond counting problems, e.g. to also consider the actual impact on evaluated systems. Law (2006) and Sawyer *et al.* (1996) apply the concept of 'downstream utility', which determines the extent to which results from usability evaluations impacts the usability of a system.

Another important part of user-based evaluations is the actual conduction, and in particular it has been argued that the test monitor role is very challenging, as the person acting this role must pay careful attention to making sure that all users get introduced to the test in the same way, and that they think aloud, creating good relations with the users and not rescuing the users (Rubin and Chisnell 2008). Häkli (2005) touches upon this issue, but further systematic studies on the matter are needed.

3. Case company

In this study, we collaborated with a small Danish software development company with just over 20 employees in their software department. The company produces web applications used within the public sector and consists of self-service solutions for citizens at the front end and administration solutions at the back end. The organisational structure is relatively flat, with the head of development leading all development teams and all other employees either develop or test the solutions and several employees take on multiple roles as, e.g. project managers and developers, etc. The company claims to follow SCRUM as the overall development method. At the onset of this study, there were no usability specialists employed and the employees in general had little to no experience in UE, which is elaborated upon in the following.

3.1. *Software development practitioners*

This subsection provides an overview of the SWPs participating in the initial and follow-up studies. They were all employed in the case company and Table 1 presents an overview of their job functions within the company and their experience with usability work in general. Most of the SWPs worked as systems developers where some also had responsibilities as project managers and SWP 2 worked as a test manager. Two of the SWPs had previous practical experience of conducting usability evaluations of which SWP 1 as part of his education attended an HCI course and in the conduction of four to five usability evaluations seven years prior to the initial study. SWP 5 had also attended an HCI course during his education and had experience from conducting a single usability evaluation 13 years prior to the initial study. SWP 2 had only theoretical knowledge on usability evaluations and she had read a single chapter on the subject during her education. Additionally, SWP 8 had only undertaken an HCI course during his education two years prior to the initial study, i.e. he had no practical experience in conducting usability evaluations. Given the SWPs job responsibilities as systems developers and project managers and their limited previous knowledge of usability evaluations, we argue that the SWPs participating in our studies were not HCI specialists.

Table 1. Overview of the SWPs' job functions within the company and experience with usability evaluations.

SWP no.	Function	Usability experience
1	Systems developer	HCI course + 4–5 evaluations
2	Test manager	Through literature
3	Project manager + systems developer	None
4	Systems developer	None
5	Systems developer	HCI course + 1 evaluation
6	Project manager + systems developer	None
7	Project manager + systems developer	None
8	Systems developer	HCI course

4. Experimental method

This paper presents two studies of which an initial study was conducted in order to assess the feasibility of the developer-driven approach in a laboratory setting at the university. The follow-up longitudinal study builds on the initial study by evaluating the developer-driven approach in a natural setting at the case company.

4.1. *Initial study*

This section presents the method applied in the initial study aiming to assess the feasibility of developer-driven usability evaluations.

4.1.1. *Participants*

4.1.1.1. Software development practitioners. All eight SWPs participated in the training given in this initial study. They received a two-day training course on how to conduct conventional user-based usability evaluations with following analysis of video data, which is elaborated in the subsection 'Initial Training' below. Five of them (SWP 1–5) planned and conducted the evaluation as well as analysed the video data.

4.1.1.2. Trainers. The two authors prepared and held a usability training course for the practitioners. The authors additionally acted as observers during the following evaluation experiment in order to provide feedback on the practitioners' performance in the role of test monitors.

4.1.1.3. External raters. Three HCI specialists acted as external raters of the problem lists produced by the SWPs, as we did not want to evaluate our own training. None of these raters had taken part in the training or the conduction of the usability evaluation and are thus considered to be unbiased.

4.1.2. *Initial training*

The authors designed and conducted a two-day training course for the SWPs. As the SWPs had little to no previous experience with usability evaluations or had not applied such methods for several years, we emphasised teaching a conventional user-based evaluation, as described in Schaffer (2007). Previous studies have shown that novice

evaluators prefer this method over non-user-based analytical evaluation methods (Ardito *et al.* 2006, Frøkjær and Hornbæk 2008) and that user-based evaluations provide valuable first-hand observations of the problems experienced (Høegh *et al.* 2006). The content of the training course followed Rubin and Chisnell (2008) and included the following topics:

- Planning
- Preparation
- Conduction
- Interpreting results
- Disseminating results

The course was held as a combination of presentation and exercises. Finally, upon completion of the course, we gave the SWPs a homework assignment in which they were asked to analyse five video clips from a previous usability evaluation of an e-mail client. We collected the resulting problem lists and gave the participants feedback on how they could improve their problem descriptions. The SWPs spent 14 hours participating in the training course (roughly 7 hours of presentations and 7 hours of exercises). Additionally, participants spent an average of 8.5 hours on the homework assignment. Thus, thus in total, the SWPs spent 22.5 hours on this initial training course.

4.1.3. *Conducting the evaluation*

After completing the initial training course, we asked the five practitioners to plan and conduct a usability evaluation of one of the system developers by the case company and to analyse the obtained video data for usability problems. This evaluation experiment was executed one month after completing the training course.

4.1.3.1. System. The system evaluated was a web application that citizens use when moving from one address to another. In that case, the municipality need information on the new address, which doctor they would like to have in their new area, etc. The system was partly developed by the software company in which the five practitioners were employed but none of the practitioners had participated in the development of the particular system.

4.1.3.2. Setting. The evaluation was conducted in the usability laboratory at the university, which consists of a test room with ceiling-mounted cameras and a microphone as well as observation and control rooms concealed by one-way mirrors. Figure 1 illustrates the layout of the usability laboratory. During each session, a test user was sitting at the table in test room 1 using the web application. Next to the user, an SWP acting as test monitor was positioned. The authors acted as observers and were sitting in the adjacent control room behind a one-way mirror during all sessions. As the authors were familiar with the technical equipment in the lab, they set up the camera positions and were responsible for starting and stopping the recordings between sessions. Figure 2 shows a snapshot of the recorded video material with a picture-in-picture set-up.

Six test users were recruited for the evaluation, all of which were representative end users of the evaluated system. The SWPs found the users and communicated with them without the involvement of the authors. None of them had used the system before.

4.1.3.3. Procedure of the evaluation. Three of the practitioners (SWPs 1, 2 and 3, see Table 1) planned and conducted the evaluation and they (plus SWPs 4 and 5) analysed the obtained video material and described the identified usability problems. The evaluation was conducted in one day with the six test users and SWPs 1, 2 and 3 acted as test monitors two times each (they took turns). The authors provided feedback on their performance in this role. This was done upon completion of each test session, as this enabled the SWPs to consider the feedback when it again was their turn to act as test monitors.

4.1.3.4. Analysis of problem lists. After completing the evaluation, the five practitioners analysed the video material individually. One of the authors, an HCI specialist with 10

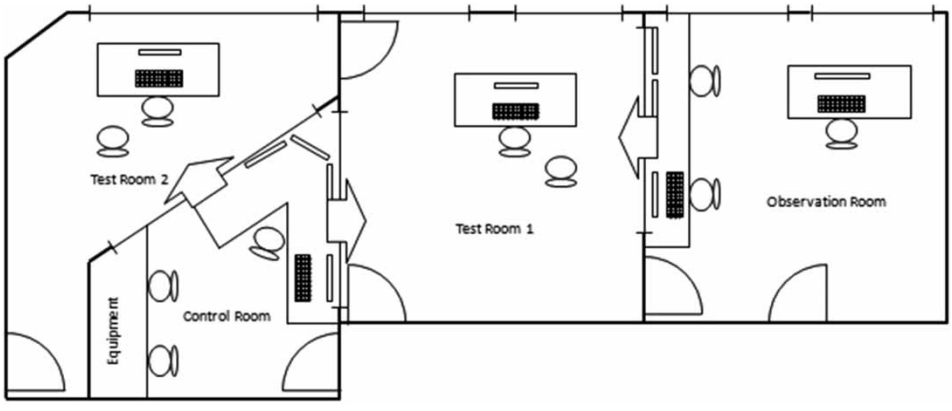

Figure 1. Layout of the usability laboratory.

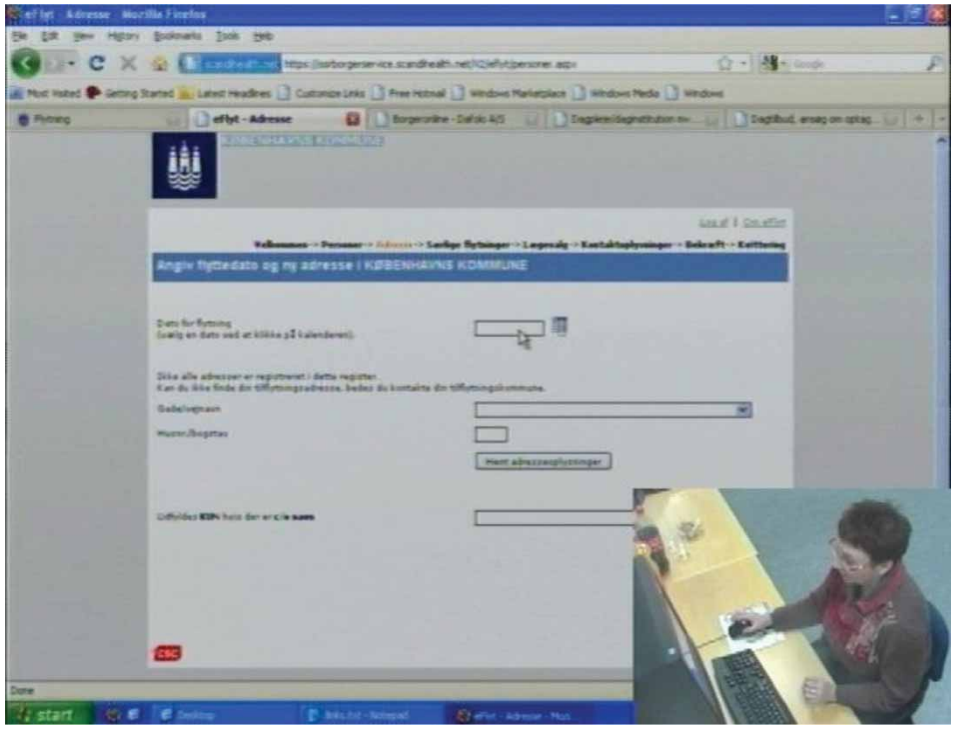

Figure 2. Snapshot of video recording.

years of practical experience in usability evaluations also analysed the same video material. The SWPs and the HCI specialist used the same template for describing problems, as we wanted to foster a consistent format. After the individual analysis, we asked the three unbiased external raters to evaluate the quality of the problem descriptions in the lists created by the five practitioners and the HCI specialist. Finally, the HCI specialist held a meeting with the five practitioners in order to merge the six individual problem lists into a total list of usability problems. The merged list served as a white list to calculate the thoroughness in identifying problems. At the same meeting, the HCI specialist held a debriefing interview with the SWPs.

4.1.4. *Measuring quality of problem descriptions*

The external raters were asked to rate the quality of the six problem lists by first reading each list and then provide a rating on a scale from 1 to 5 (1 = 'Not fulfilled′, 2 = 'Scarcely fulfilled′, 3 = 'Partially fulfilled′, 4 = 'Almost fulfilled′ and 5 = 'Fulfilled′). These ratings are given on the following attributes, which extend the quality considerations mentioned in the related work, see Capra (2006):

(1) *Be clear and precise while avoiding wordiness and jargon:* Define terms that are used. Be concrete, not vague. Be practical, not theoretical. Use descriptions that non-HCI people will appreciate. Avoid so much detail that no one will want to read the description.
(2) *Describe the impact and severity of the problem:* This includes business effects (support costs, time loss, etc.), impact on the user's task and importance of the task. Describe how often the problem will occur and system components that are affected or involved.
(3) *Support your findings with data:* Examples: how many users experienced the problem and how often; task attempts, time and success/failure; critical incident descriptions and other objective data, both quantitative and qualitative. Provide traceability of the problem to observed data.
(4) *Describe the cause of the problem:* This includes context such as the interaction architecture and the user's task. Describe the main usability issue involved in the problem. Avoid guessing about the problem cause or user's thoughts.
(5) *Describe observed user actions:* This includes specific examples from the study, such as the user's navigation flow through the system, user's subjective reactions, screenshots and task success/failure. Mention whether the problem was user-reported or experimenter observed.

Finally, the external raters were asked to provide a qualitative assessment of each list, i.e. to relate arguments of the ratings given to examples from the problem lists.

4.2. Follow-up longitudinal study

This section describes the procedure applied in the follow-up study of the developer-driven approach. This was a longitudinal study aiming to evaluate the downstream utility of the approach and was conducted five months after the initial study. Downstream utility is a measure of the impact that evaluation results have on the usability of a software system (Sawyer *et al.* 1996, Law 2006), which is further elaborated in the subsection 'Measuring Downstream Utility' below. The idea behind the method applied in the follow-up study was to let the SWPs evaluate two versions of the same system in order to assess its usability before and after redesigning it. They evaluated the first version of the system after which they spent three months fixing the problems identified. The time span of three months was selected so that the practitioners had sufficient time to fix the problems. After three months, the second usability test was conducted to determine the effectiveness of the fixes, i.e. the downstream utility.

4.2.1. Follow-up training course

The conventional usability evaluation method taught during the initial training necessitates traversing several hours of video data, which requires a considerable amount of resources. As mentioned in the introduction, then small software development companies in particular experience the barrier of high resource demands in connection with usability evaluations. For this reason, we also chose to train the SWPs in applying instant data analysis (IDA), as this method is not based on reviewing video data. IDA is conducted immediately after the final test session and includes the following steps (cf. Kjeldskov *et al.* 2004):

(1) *Brainstorm:* The test monitor and data loggers participating in the test identify 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 follow-up course in IDA was held by the authors two months after the initial training on conventional usability evaluation, which was held as part of the initial experiment. A combination of presentations and exercises was also applied in this course, which had a duration of seven hours. When combining the initial and follow-up training courses, the SWPs received a total of 30 hours of training on conducting usability evaluations.

4.2.2. Participants

4.2.2.1. Software development practitioners. Three SWPs participated in the follow-up study (SWPs 6, 7 and 8, see Table 1). As was the case in the initial study, the SWPs were asked to plan and conduct the two evaluations as well as identify usability problems. This time, however, problem identification was done based on IDA rather than conventional video data analysis.

4.2.2.2. HCI specialists. Three HCI specialists analysed the video material obtained from both tests in order to compare performance to that of the SWPs. Two of these were external HCI specialists employed in industry and the third was an HCI researcher (the same person who participated in the initial study who had 10 years of usability evaluation experience). The two external specialists were newly graduated master students and had each over 100 hours of experience in conducting usability evaluations. The external specialists had not otherwise taken part in the experiment. None of the specialists had previous experience in the domain in which the evaluated system is used.

4.2.3. Conducting the evaluations

4.2.3.1. System. The system evaluated was a web application designed to register and apply for wage subsidies by administrative staff within companies. Wage subsidy applications are typically filled out by the administrative staff and then sent to the municipality. The municipality then provides companies with subsidies for the employees enrolled in such settlements. The system was developed by the small software company in which the practitioners were employed and consisted of the following two parts:

(1) A stepwise wizard in which the data would be entered.
(2) A pdf form shown as a confirmation at the end of the wizard in which users could edit previously entered data.

4.2.3.2. Setting. Figure 3 shows the setting applied in the evaluations, which in the follow-up study were conducted in an office at the case company. We configured video capture software in order to record how test users interacted with the system and a webcam captured users' faces. An external microphone was used to record the audio. Figure 2 illustrates the recordings made in the initial study, which are similar to the ones made in this follow-up study. In essence, the audio and video recordings were unnecessary for the SWPs conducting IDA, however, the purpose of recording the sessions was to enable HCI specialists to perform video analysis at a later point, which we later used for comparison purposes.

During each session, a test user was sitting at a table in the office solving the pre-defined tasks by using the system.

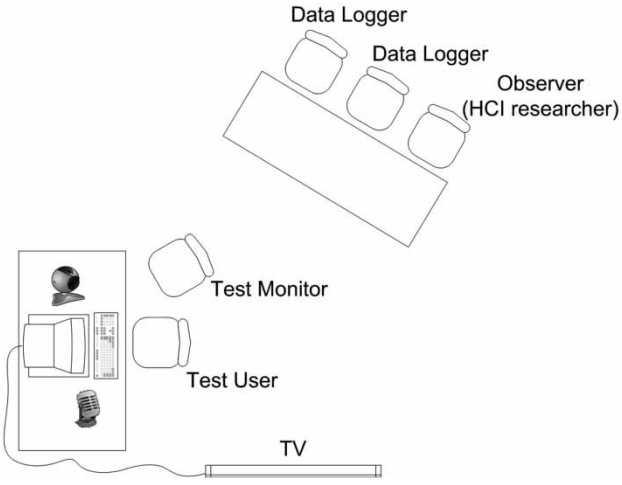

Figure 3. Overview of test settings.

One practitioner acted as the 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 metres behind the test user in order not to interfere.

A total of seven test users participated in the two evaluations, all of which were recruited by the practitioners without the involvement of the authors. Three test users participated in the first evaluation and four other users in the second. The test users were employed as administrative staff within different companies and all had experience in applying for wage subsidies. None of them had used the system before.

4.2.3.3. Procedure of the two evaluations. The three SWPs participating in the follow-up study planned and conducted the two evaluations including finding the test users as well as defining the three tasks given to them. The same three tasks were given in both evaluations. As mentioned previously, the two evaluations were conducted three months apart. The SWPs also distributed the roles of test monitors and data loggers between them.

Each of the two evaluations was conducted in one day, as prescribed by the IDA method (Kjeldskov *et al.* 2004). For each evaluation session, a user would be introduced to the procedure and the system by the test monitor. Then the user was asked to solve the three tasks one by 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 IDA to create a list of usability problems. During both analysis sessions, one of the data loggers acted as facilitator by noting and organising the identified usability problems on a whiteboard.

4.2.3.4. Improving the system. The list of usability problems identified after 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 with the purpose of discussing what problems to fix and to discuss redesign proposals. This was followed by three months of development, which was mainly done by SWP 8, who did not have any project management responsibilities. During the three months of developing a new version of the system, the SWPs held weekly meetings to discuss how they progressed.

4.2.3.5. Interviews. After the second evaluation, the three SWPs were interviewed by one of the authors. The interview was semi-structured and to facilitate discussion, this was conducted with all SWPs present at once. The aim of the interview was to uncover factors influencing how the SWPs prioritised fixing the identified usability problems and reasons of why some problems recurred in the second evaluation. Audio recordings from the interview were transcribed by one of the authors and analysed using grounded theory at open-coding level (Strauss and Corbin 1998).

4.2.3.6. Analysis of problem lists. The three HCI specialists (one of the authors and two external) analysed video recordings from the two evaluations. To reduce ordering bias, videos were analysed in random sequence. As done in the initial study, the specialists applied the same document template as the practitioners for describing usability problems. The severity of each problem was categorised as either cosmetic, serious or critical, corresponding to the categorisations applied by the practitioners. Analysis was first done individually where each specialist created two problem lists; one for the first version of the system and another for the second revised version. After the individual analysis, all specialists held a meeting where they merged individual lists into two lists consisting of all identified problems for the first version of the system and a similar list for the second. The specialists negotiated the severity of problems until an agreement was reached. As the final step, the two lists created by the SWPs through the IDA sessions were merged into the lists created by the specialists. In case of identical problems, the severity ratings given by the SWPs were overridden by that given by the specialists. Severity ratings on problems uniquely identified by the SWPs were not altered.

4.2.4. Measuring downstream utility

This section describes the measurements utilised in our analysis to determine the level of downstream utility of the developer-driven usability evaluations examined in the longitudinal follow-up study. Downstream utility is a measure of the extent to which results from usability evaluations impact the usability of a software system. Throughout the literature, downstream utility is measured in terms of

the committed impact ratio (CIR) and completed-to-date impact ratio (CDIR) (cf. Sawyer *et al.* 1996, Law 2006). CIR denotes the extent to which a development team commits to fixing usability problems before the implementation of a redesign takes place. CIR is calculated as follows:

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

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

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

5. Findings

We start by presenting findings related to test monitor performance, which are derived from the initial study in laboratory settings. We then present results on thoroughness and problem agreement, which were measured in both studies. This is followed by measurements related to the quality of problem descriptions, which were derived from the initial study. Finally, results on downstream utility from the follow-up study are presented.

5.1. Test monitor performance

Three of the five practitioners (SWPs 1, 2 and 3, see Table 1) took turns in acting as test monitors during the evaluation in the initial study. In general, we found that all three of these practitioners ran the sessions according to plan as the test users started working on all scheduled tasks within the time frame. Below we describe the challenges experienced.

5.1.1. Reading the orientation script

The test monitors role was to introduce the users to the test by reading a printed orientation script. During the sessions, we observed that all three SWPs experienced difficulties in this respect. SWP 1 read the text aloud for the users, but in the following feedback session he mentioned that: 'It felt weird to read the text aloud when a stranger is sitting next to you.' Similar comments were also made by SWPs 2 and 3 who instead of reading the text were using their own wording and SWP 3 mentioned that: 'I wasn't reading the text directly, I explained each of the sections. It felt awkward to read it aloud.'

5.1.2. Relation to test users

In general, we found that all three SWPs created relaxed conditions for the test users. SWP 3, for instance, started his sessions by explaining to the users that 'we are working on artificial data, so the situation may seem a bit strange.' This may have a calming effect as it makes the users feel that he or she should not worry about the unnatural situation posed by the laboratory setting.

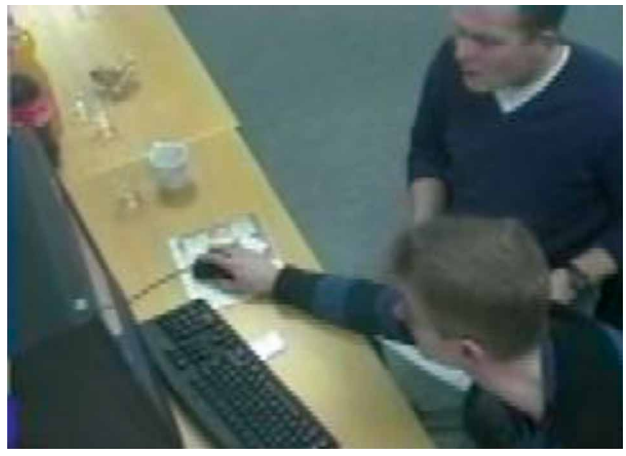

Figure 4. Test monitor leaning in over the user and taking control.

However, we also observed less relaxing moments, e.g. in one of the sessions the user had completed the first task and the test monitor (in this case SWP 1) wanted to prepare the system for the second task. In doing so, he leaned over the user, took the mouse out of the users' hands and started interacting with the system without explaining what was going on. Judging by the body language this left the user in a state of bafflement, as he quickly moved back on the chair, see Figure 4.

5.1.3. Making users think aloud

All three SWPs experienced problems in making the test users think aloud, which was observed as pauses in the narratives and the types of questions asked. In one of the feedback sessions, SWP 1 mentioned that it felt like he had been talking too much during the test: 'It [probing the users] felt unnatural and I thought I was talking too much.' However, with him acting as test monitor we observed several pauses in the narrative. When confronted with this, he replied that:

> . . . I didn't find the pauses strong enough to interfere and I think I interpreted the situation in way where I believe the users need a break in order to get an overview. I didn't interfere because I thought the user was reading the text.

SWPs 2 and 3 were better at making users think aloud.

Observations of SWP 2's performance revealed fewer pauses compared with SWP 1 and we observed that whenever users did not fill in certain input fields in the system, she asked why, which gave way to valuable input on usability problems. She was, however, unsure of her own performance as she in one of the feedback sessions mentioned that: '. . . I was in doubt of whether I made the user comment in the right situations or if some of the things should have been brought up in the following interview instead'.

Observations of SWP 3's performance also revealed fewer pauses compared with SWP 1. We observed that he in the first half of the test sessions was active in making the users think aloud, as he asked follow-up questions to missing input like SWP 2. However, in the second half of the sessions he stopped asking questions and there were considerable pauses in the narratives of the users.

Another observation we made was that when he was active in making the users think aloud he focused on asking the users to state what they were doing which resulted in the users saying 'I'm now pushing this button,' 'Now I type the address', etc. Thus, he was not asking users why they were doing it or why they hesitated in certain areas of the interface.

5.1.4. *Rescuing test users*

We observed multiple incidents where SWPs 2 and 3 rescued the test users too early. As an example, SWP 2 had a user that was entering data in the wrong input field and she then explicitly gave the correct answer to the user and furthermore explained the purpose of the input fields.

5.2. *Thoroughness*

In both studies, we examined the proportion of usability problems found by the SWPs out of the total set of identified problems and compared this to the thoroughness of HCI specialists. In the following, we start by presenting the thoroughness findings observed in the initial study after which we present the findings from the follow-up longitudinal study.

5.2.1. *Initial study*

Figure 5 presents an overview of the number of problems identified by each of the five SWPs and the HCI specialist. Results show that a total of 50 usability problems were identified of which 12 were critical, 19 serious and 19 cosmetic. Figure 1 shows that the HCI specialist identified 31 of the problems (62%) and the SWPs identified between 14 (28%) and 33 (66%), the mean being 24.2 (SD = 8.1) or 48%. The 95% confidence interval spans over [19.3; 30.7]. Findings also show that SWPs identified an average of 6.8 (SD = 2.6) of the critical problems (57%) and in comparison the HCI specialist identified 6 (50%). In case of the serious problems, SWPs found 10 (SD = 3.9) on average (53%), whereas the HCI specialist found 12 serious problems (63%). Considering the cosmetic problems, we found an average thoroughness of 7.4 (SD = 3.2) for the SWPs (39%) while the HCI specialist found 13 (68%). Figure 5 also reveals that SWPs 1 and 2 performed on par with the HCI specialist while SWPs 3, 4 and 5 had a lower thoroughness.

In practice it can be too resource demanding to utilise five evaluators in analysis of usability data, and in the following we study the thoroughness of each pair of SWPs. Figure 6 provides an overview of the number of problems identified by all pairs of SWPs. All pairs identified an average of 35.7 (SD = 5.2) of all problems (71%) and the 95% confidence interval is [33.5; 38.9]. SWP 1 and SWP 5 was the pair that identified most problems (86%) and SWP 3 and SWP 5 identified fewest (52%). In comparison, the HCI specialist identified 62%.

It should be mentioned that the pair SWPs 1 and 5 had the highest level of thoroughness, but that these two SWPs had previous practical experience with conducting usability evaluations, see Table 1. For this reason, all pairs consisting of either SWP 1 or 5 were removed. When this is done, we see that the average number of identified problems drops from 35.7 (SD = 5.2) to 33.3 (SD = 4), which amounts to 67% of all problems.

5.2.2. *Follow-up longitudinal study*

Figure 7 provides an overview of the number of problems identified by the SWPs and HCI specialists in the two evaluations conducted in the follow-up study. The SWPs applied IDA while the specialists based their analysis on video data. Taken together, the SWPs and specialists identified 41 problems in the first version of the system (evaluation 1)

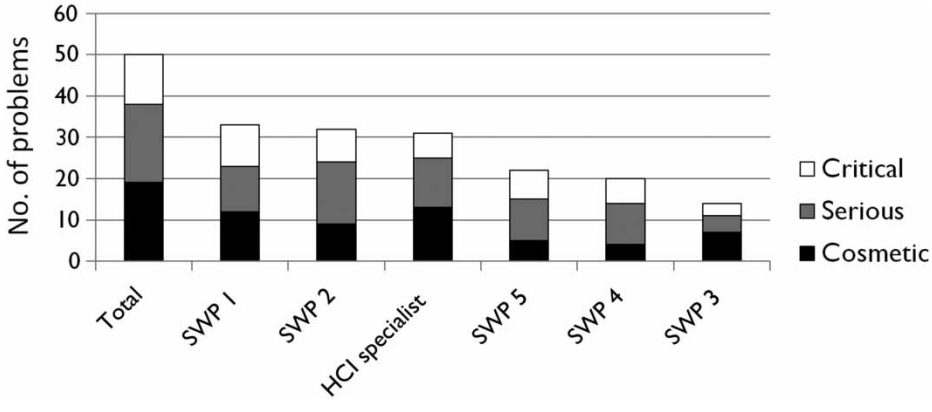

Figure 5. Overview of the number of problems identified by the SWPs and the HCI specialist.

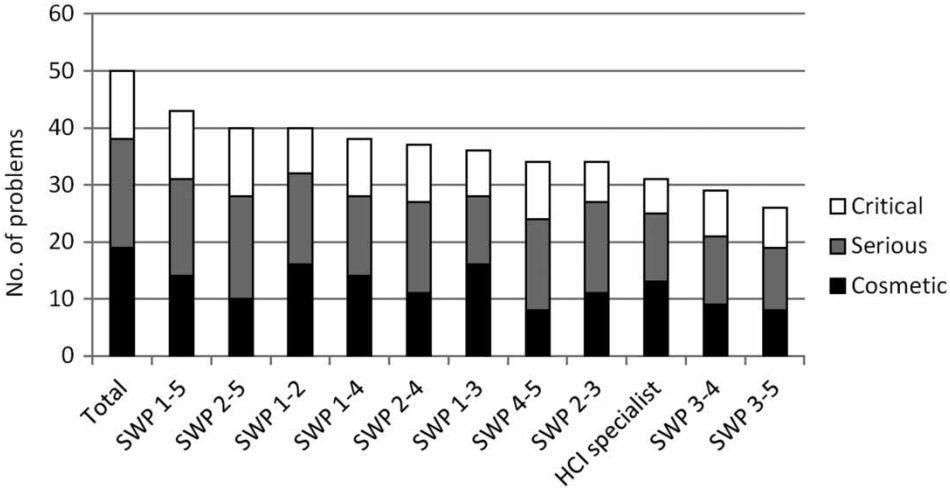

Figure 6. Overview of the number of problems identified by all pairs of practitioners.

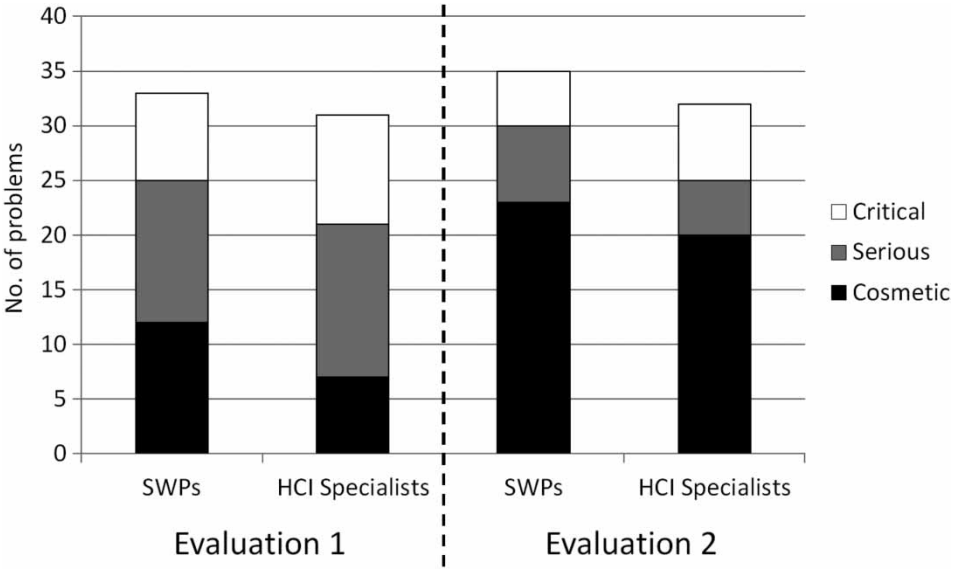

Figure 7. Overview of the number of problems identified by the SWPs and the HCI specialists in both evaluations.

and 44 in the second (evaluation 2). This gives a 95% confidence interval of [40; 44.9]. The practitioners identified 81% and 80% of all problems in evaluations 1 and 2, respectively, while the specialists identified 76% and 73%. A Fisher's exact test reveals no significant differences between the thoroughness of the SWPs and specialists in these two evaluations ($df_{Evaluation\ 1} = 1$, $p_{Evaluation\ 1} > 0.7$ and $df_{Evaluation\ 2} = 1$, $p_{Evaluation\ 2} > 0.6$).

In both evaluations, the SWPs identified fewer critical but more cosmetic problems than the specialists. Considering the serious problems, the SWPs identified one less in the first evaluation and two more in the second.

Findings also reveal that the number of critical and serious problems was almost halved in the second version of the system as the count decreased from 26 to 15. However, the number of cosmetic problems was doubled as they increased from 15 to 29. On average, the specialists identified 48% and 41.7% of all problems found in evaluations 1 and 2, respectively. Due to the plenary nature of the IDA session, such an overview cannot be made for the SWPs.

5.3. *Problem agreement*

We calculated the evaluator effect based on Hertzum and Jacobsen (2001) and found an any-two agreement of 38% between the SWPs and the HCI specialist in the initial study. The internal any-two agreement between SWPs in this study was also 38%.

Due to the plenary nature of IDA, we cannot calculate the any-two agreement between the SWPs in the follow-up study. We can, however, calculate the agreement between the list of usability problems found by the SWPs in total and the total list of problems found by the specialists. In the first evaluation, we found that 23 of the total 41 problems were

identified by the SWPs as well as the specialists. This leads to a problem agreement of 56%. In the second evaluation, the two groups of participants also had 23 of the 44 problems in common, which gives an agreement of 52%. Thus, in the follow-up study the average agreement is 54% (SD = 2.8).

5.3.1. *Differences in severity categorisations*

In the follow-up study, we found that the practitioners and specialists had an agreement on 23 problems in the first evaluation of which 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%) where the practitioners once more provided lower ratings than the specialists. Thus, we found an average disagreement of 46% in severity ratings.

5.3.2. *Uniquely identified problems*

In the follow-up study, we found that 22 problems were identified by the practitioners only, of which 2 are critical, 3 serious and 17 cosmetic. In the following, we provide an example of one of the serious problems. 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 tests, 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 test. Other similar problems were identified by the practitioners but not by the usability specialists.

5.4. *Quality of problem descriptions*

This subsection describes the SWPs' ability to describe usability problems according to the five quality attributes of clarity, impact, data support, cause and user actions derived from Capra (2006). These observations are derived from the initial laboratory study.

Table 2 provides an overview of the median quality ratings given by the three external raters. Higher ratings indicate a higher level of fulfilment according to the quality attributes (on a one- to five-point Likert scale). The table shows that problem descriptions written by SWPs 1 and 5, who received the median scores of 4 and 3, respectively, described their usability problems with a quality comparable to that of the HCI specialist (median = 4). The other three SWPs scored a lower median rating of 2. Additionally,

Table 2. Median quality ratings given by the three external raters to the problem lists written by the SWPs and the HCI specialist.

Participant	Clarity	Impact	Data	Cause	Actions	Overall median
SWP 1	4	3	4	4	4	**4**
SWP 2	2	2	2	2	2	**2**
SWP 3	2	2	2	2	1	**2**
SWP 4	3	2	2	2	2	**2**
SWP 5	4	2	3	3	3	**3**
Overall median	**3**	**2**	**2**	**2**	**2**	**2**
HCI specialist	4	3	4	3	5	**4**

the table indicates that practitioners are better at being clear and precise (clarity) in their problem lists than any of the other attributes, which is elaborated upon below along with examples of the qualitative comments made by the external raters.

5.4.1. *Clarity*

Table 2 shows that the SWPs were better at fulfilling the clarity attribute than any of the other attributes, as they scored an overall median of 3. In comparison, the HCI specialist received the median rating of 4 by the external raters. This was also the case for SWPs 1 and 5. As an example on the qualitative comments given, one of the raters mentioned that SWP 5's list provided 'Good insights in the problems experienced'. The list created by SWP 2 received rather different ratings on the clarity attribute where one rater stated that: 'Descriptions are so short that it is sometimes impossible to understand the problem' while another mentioned that: 'The list in general is short and clear without use of HCI-jargon.' Thus, there were diverging opinions between raters on how long problem descriptions should be. In relation to this, we also found diverging opinions on the ratings given to the list written by the HCI specialist where one rater found that the list had 'good and detailed descriptions' while another stated that the list was 'too wordy'. SWPs 2, 3 and 4 scored the lowest median ratings on this attribute where one rater mentioned the following about SWP 3's list: 'Extremely short and imprecise descriptions. Actually the descriptions are so poor that you in most cases cannot find out what the problem is.'

5.4.2. *Impact*

Table 2 also shows that lower median ratings were given with respect to the impact attribute compared with clarity, which is the case for both the SWP and specialist descriptions. SWPs got an overall median of 2 and the HCI specialist 3. SWP 1 performed on par with the specialist on this matter and got a higher median rating than the remaining SWPs. One of the external raters commented that practitioners in some problems describe the impact on the user's task

but other elements, such as business effects and affected system components, are left unmentioned. This is also the case for descriptions provided by the HCI specialist. Rater 3 mentioned that impact is not explicitly mentioned in the problem descriptions.

5.4.3. *Data support*

In terms of providing data support for problem descriptions the SWPs received an overall median rating of 2 and again SWPs 1 and 5 scored highest (4 and 3, respectively). In comparison, the specialist received the median rating of 4 on this quality attribute. One of the raters commented that practitioners in general describe how many test users that experience given problems and that they in certain descriptions state whether or not the task was a success or a failure. This rater also found that objective data, such as traceability information, is left unmentioned. Another mentioned that: 'Many problems are not clearly connected to observations,' thus this rater found that practitioners did not always consider objective data. The same rater additionally mentioned that practitioners made use of vague statements such as 'The user does not understand' or 'the user is in doubt,' statements which are of a speculative nature. However, the practitioners did describe how many test users that experienced the problems and whether or not the tasks were completed, which is similar to the information provided by the HCI specialist. Additionally, it was commented that the HCI specialist provided 'good descriptions of the critical incidents'.

5.4.4. *Problem cause*

On this attribute, an overall median rating of 2 was given on SWPs' descriptions where the HCI specialist received a median of 3. SWPs 1 and 5 once more scored higher median ratings than the other three (4 and 3, respectively). One of the external raters mentioned the following about the descriptions provided by SWP 1: 'The list is ok with good descriptions that to a great extent describe causes,' which was agreed upon by another rater. The third rater, however, found that this practitioner was guessing on the users' thoughts and the cause of the problem in some of his descriptions. Similar diverging opinions were observed in practitioner 5's list on this matter. Practitioners 2, 3 and 4 were given the lowest ratings in which case all three raters agree that no causes or arguments were provided.

5.4.5. *User actions*

Finally, Table 2 shows that SWPs and the specialist received median ratings of 2 and 5, respectively, in terms of describing user actions. Two of the raters mentioned that several of the descriptions provided examples on users' navigational flow, but that reactions are sometimes described implicitly by stating that users 'are in doubt' or 'overlooks' certain elements in the interface. However, according to one of the raters, SWPs 2 and 3 do not describe user reactions at all. Yet again, SWPs 1 and 5 scored the highest ratings compared with the other SWPs, where they received medians of 4 and 3, respectively. Two raters found that the descriptions written by the specialist contained detailed information on users' navigational flow and reactions.

Summarising on the quality of the problem descriptions, we found that SWPs were better at fulfilling the clarity attribute than the other four. In addition, we saw that SWPs 1 and 5 scored the highest median ratings and that they both performed on par with the HCI specialist regarding the attributes of clarity, data support and problem cause. SWP 1 also received the same rating as the specialist with respect to the impact attribute. In general, SWP 1 provided the same quality in problem descriptions as the HCI specialist.

5.5. *Downstream utility*

This subsection presents findings related to the downstream utility, which are obtained from the longitudinal follow-up study.

5.5.1. *Committed impact ratio*

After completing evaluation 1 in the follow-up study, the SWPs had their own problem list obtained by applying IDA. The problem list made by the specialists was not available before starting the redesign and implementation. Thus, in the following, we apply the 33 problems found by the SWPs only as the total number of identified problems and *not* the 41 problems identified in total when including problems identified by the specialists. Before starting the implementation, the SWPs committed to fix 20 problems, which results in the following CIR:

$$\text{CIR} = \frac{20}{33} \times 100 = 61\%.$$

In the interview conducted at the end of the experiment, we asked the SWPs of what factors had influenced the CIR, which were derived from the existing literature: Severity ratings, frequency, length of problem descriptions and resource requirements (cf. Law 2006).

We found that the amount of resources going into fixing a usability problem was one of the main factors influencing CIR. As an example, one of the SWPs 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 SWP followed up 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.' This indicates that resource requirements had higher influence than severity ratings.

Frequency measured in terms of the number of users experiencing a problem was not an influential factor, an SWP 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 emphasised over those experienced by a single test user in this case. However, frequency may be influential with a larger user base in evaluations.

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

Additionally, we found that the SWPs did not find the length of problems descriptions influential on their prioritisation of fixing problems. 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 [IDA] is conducted immediately after the sessions so we do remember them.'

We uncovered an additional factor regarding coherence to other systems in the company portfolio. As an example, the case 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 prioritised, 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"'. For this reason, the SWPs only prioritised fixing problems related to the part of the wage subsidy system containing the stepwise wizard and not the editable pdf form.

In summary, the SWPs 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.

5.5.2. *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:

$$\text{CDIR} = \frac{21}{33} \times 100 = 64\%.$$

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

In case of the other eight recurring problems, the SWPs mentioned that they tried to redesign and implement fixes for five of these, but that the 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 interview also revealed that one of the recurring problems was not accepted by the SWPs after the first evaluation and one of them mentioned: 'Well you could say that we should have taken this problem more seriously after the first test, 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 prioritised by the sales department. As an example, one of them 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.'

Summarising on the above, we found that the SWPs tried to fix most of the 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 prioritised after its presence in the second.

6. Discussion

This section presents a discussion of our findings, which we compare to related work and the discussion is structured around the four research questions.

6.1. *Identification of usability problems*

The first research question regarded the extent to which software practitioners with minimal training in usability evaluations are able to identify usability problems. This is related to the thoroughness of evaluators in identifying problems but also the reliability in doing so. The latter was measured through the any-two agreement. In the following two subsections, we discuss these topics.

6.1.1. *Practitioners outperform specialists in thoroughness*

Our findings reveal that the SWPs were able to identify a considerable amount of usability problems in the initial study as well as in the follow-up study. The initial study showed that each practitioner on average identified 48% of

all problems and that the usability specialist in comparison found 62%. Arguably, the practitioners perform well below the specialist in this case, but results from that study also show that a pair of practitioners had an average thoroughness of 71%, i.e. a pair of practitioners was able to outperform one usability specialist in this respect when conducting traditional video-based analysis. In the follow-up study, we found that three practitioners conducting IDA had a thoroughness of 80% and that three specialists conducting conventional video-based analysis identified 74%, thus the practitioners outperformed the specialists.

In general, related work report a lower thoroughness than the one found in our experiment. The study presented in Wright and Monk (1991) show that each student team identified 33% of all problems on average. In the study conducted by Koutsabasis *et al*. (2007), it was found that students applying the user-based method were able to identify 24% of all problems on average. In Frøkjær and Lárusdòttir (1999), it is shown that students revealed 18% of all problems, whereas the level of thoroughness reported in Ardito *et al.* (2006) is lower as the students applying the user-based method identified a mean of 11%. The three studies presented in Skov and Stage (2004), Skov and Stage (2008), and Skov and Stage (2009) compare student performance to that of specialists and show that students identified a mean of 37% of the problems identified by specialists.

Differing motivational factors can explain part of the variations between related work and our study. In a competitive market, SWPs are dependent on product revenue, which is not the case for university students, which are applied as the empirical basis in related work.

We also found that the three practitioners in the follow-up study had a higher thoroughness than the three specialists, but the difference is not statistically significant. Could this be attributed to the poor performance of the specialists? We believe not as the thoroughness of each usability specialist in our study on average revealed 45% of all problems. This is comparable to the thoroughness presented in Jacobsen *et al.* where four specialists conducting video-based analysis identified an average of 52% of all problems.

In our studies, it can be argued that the specialists share the properties of external consultants, as they were employed in other companies and universities than the case company, which developed the evaluated systems. Thus, the practitioners' level of domain knowledge was higher than that of the specialists. According to Bruce and Morris (1994), an inherent problem in applying external consultants is their lack of domain knowledge. The importance of domain knowledge in usability evaluations is supported in other studies, e.g. Nielsen's (1992) study of usability specialists, non-specialists and double experts. Findings from that study show that usability specialists found more problems using heuristic evaluation than non-specialists while the double experts found most problems (Nielsen 1992). Additionally, Følstad and Hornbæk conducted a study in which a group of end users acted as domain experts in the conduction of Coorporative Usability Evaluations (Følstad and Hornbæk 2010). That study shows that evaluation output was enriched by including domain experts in the interpretation phase, as they provided additional insights in identified problems and helped in uncovering a considerable amount of new problems (Følstad and Hornbæk 2010). Thus, these studies show that domain knowledge plays a key role in the identification of usability problems. This indicates an advantage of the barefoot usability evaluation approach over, e.g. outsourcing approaches where usability specialists are distant from the team that develops the software, and as a consequence lack domain knowledge.

6.1.1.1. Comparable any-two agreement. The any-two agreement is an expression of the average proportion of problems in common between all pairs of evaluators and is a measure of the reliability of usability evaluation methods (Hertzum and Jacobsen 2001). In the initial study, we found that the practitioners had an any-two agreement of 38% based on the traditional video-based analysis. Due to the plenary nature of IDA, it is not possible to derive this measure for the practitioners in the follow-up study. In comparison, however, the three specialists doing video-based analysis in the follow-up study had an average any-two agreement of 44%, which is comparable to that of the practitioners in the initial study.

The study presented in Hertzum and Jacobsen (2001) shows that the any-two agreement between usability specialists ranges from 6% up to 42%. Thus, the performance of the practitioners in our studies is at the higher end of the scale and comparable to that of specialists, i.e. our findings indicate that the reliability of novice practitioners is on par with specialists.

6.2. *Differences in identified problems*

The second research question relates to how problems identified by the software practitioners differ from those found by HCI specialists. In the following two subsections, we discuss these differences in terms of severity ratings and quality of problem descriptions.

6.2.1. Subjective bias in severity ratings

In the follow-up study, we found considerable disagreements between the severity ratings given by the practitioners and specialists. We found disagreements in 46% of all problems where the practitioners consistently gave lower ratings, e.g. the practitioners would rate a problem as cosmetic where specialists would rate the same problem as serious. This finding could indicate a potential downside to letting the practitioners test their own systems, as they may be subjectively biased. This is supported within the existing literature establishing that development teams for objectivity reasons should not test their own designs (Rubin and Chisnell 2008). Although objectivity could be questioned,

we found that the practitioners uncovered 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, findings indicate that subjective bias has a higher influence on severity ratings than on the number of problems identified.

6.2.2. *The quality of clarity*

Most of the practitioners were unable to fulfil the quality attributes in their problem descriptions to the same degree as the HCI specialist. Two of the software practitioners, however, provided a quality comparable to that of the specialist. An explanation for the performance of these two software practitioners could be their previous experience with usability evaluations. Still, however, the average quality of the practitioner descriptions corresponds to findings in Skov and Stage (2004), Skov and Stage (2008), and Skov and Stage (2009) in which it is reported that qualitative aspects of the problem descriptions written by students is poorer than that of HCI specialists. We found that practitioners were better at providing clear and precise problem descriptions than they were at describing the impact, cause, user actions and providing data support for observations. The findings in Howarth (2007) and Howarth *et al.* (2007) are different as that the students in those studies received highest ratings on the attribute related to description of user actions.

A reason for the observed differences in problem description quality may be located in the fact that some of the SWPs in our study are used to provide code comments in their software. During one of the debriefing interviews in the initial study a practitioner mentioned: 'I find it important to write understandable code comments because it's easier to get back into the code if you've had one or two weeks of vacation.' Thus, clarity as a quality attribute is important to industry practitioners and perhaps more important than in the case of students which could indicate a difference between these two types of participants.

6.3. *Challenged by the test monitor role*

In this section, we discuss findings related to our third research question on how software practitioners conduct usability evaluations in comparison to best-practice. In our study, we emphasised their performance as test monitors.

Findings from this study suggest that SWPs experience a range of challenges when conducting usability evaluations. Acting as test monitors we found that they felt awkward reading the printed orientation script out loud, as this was more artificial. Additionally, we observed that practitioners experienced problems in making the test users think aloud. This was observed through pauses in the narratives made by one of the practitioners and this practitioner also mentioned that it felt unnatural to constantly probe the users, i.e. he felt that he was talking too much. Another practitioners kept asking test users 'what' questions instead of 'why' questions, which can be problematic when conducting a formative evaluation as this provides fewer insights into the intentions of the users. Furthermore, we observed multiple incidents where two practitioners rescued the test users too early. On the positive side, we found that all three practitioners who acted as test monitors created relaxed conditions for the test users and that the test sessions ran according to plan, as users started working on all scheduled tasks within the time frame.

Considering related work, we have identified a single other study that describes observations on the performance of SWPs in the role of test monitors (Häkli 2005). In that study, it is briefly mentioned that test sessions went 'quite nicely' although they were rather unmanaged as practitioners were unable to keep the test on track. This finding contradicts our observations, which may be explained by differences in preparation time. The practitioners in Häkli (2005) were asked to plan and execute an evaluation in one day, whereas in our case the practitioners had a month to plan the evaluation. Häkli (2005) also mentions that practitioners experienced difficulties in making users think aloud which is in line with our findings.

6.4. *Most problems get fixed*

The fourth research question is on the level of 'downstream utility' of usability evaluations conducted by software practitioners. We have applied the notion of downstream utility as a measure of the extent to which results from usability evaluations impact the usability of a software system (Sawyer *et al.* 1996, Law 2006). We found that the practitioners committed to fixing most of the identified problems and that they prioritised these based on the factors of resource requirements and coherence to other systems. Additionally, the practitioners managed to eliminate most of the problems. Findings of downstream utility is also examined in other experiments utilising user-based tests, in these, however, usability practices were already established in the case companies, as specialists were involved in evaluation and redesign of the systems. Medlock *et al.* (2002) revealed a downstream utility of 97% by applying the Rapid Iterative Testing and Evaluation (RITE) method, which is higher than that found in our study. In Hertzum (2006), the average downstream utility is 65%, which was obtained through the conduction of five user-based tests. In Law (2006), usability specialists conducted a similar user-based usability evaluation based on video analysis. In that study, the downstream utility is 38.3%. Thus, the downstream utility of 64% found within our study resembles that presented in Hertzum (2006), which was obtained from a company with established usability practices and 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 tests, e.g. writing new code, fixing functionality problems and project management responsibilities.

These findings, combined with the fact that the practitioners identified a considerable amount of problems, indicate that the barefoot usability evaluations caused practitioners to accept results from usability evaluations as well as prioritise fixing problems, which deviates from the typical developer mindset described in the literature (cf. Bak *et al.* 2008, Ardito *et al.* 2011). This finding may be explained by the awareness that follows from the direct observation of users interacting with the software application, as this provides first-hand insights into the usability problems experienced by the users (cf. Høegh *et al.* 2006).

Finally, although the practitioners in the barefoot approach managed to eliminate most of the problems found in the initial version of the system, it was also found that the second version introduced a considerable amount of new problems. This behaviour is recognised by Nielsen who argues that design and evaluation should be conducted over several iterations, as a new design may introduce new usability problems (Nielsen 1993).The number of new problems could be reduced if practitioners not only received training in evaluation, but also in interaction design. As Wixon (2003) points out, then it is equally important to tell the practitioners what to do and not just what is wrong within an interface. 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.

6.5. Cost effectiveness

The fourth research question on the level of downstream utility is an external metric determining the actual impact on the system, i.e. evaluation performance is viewed in a broader organisational context compared to internal metrics, such as thoroughness and test monitor performance. In relation to external metrics, we also find it relevant to initiate a discussion on cost effectiveness.

We found no statistically significant difference between the practitioners and the usability specialists in terms of thoroughness after receiving 30 hours of training. This shows that such practitioners can obtain considerable competences in what may seem to be a short time frame. On the other hand, it may be difficult to overcome the barrier of high resource demands when each practitioner has to spend 30 hours on training. Thus to avoid this initial overhead of training, it may be more feasible to, e.g. apply an outsourcing approach where an external usability specialist with the right competences conducts the evaluations. In the long run, however, it can be argued that barefoot usability evaluations would require less resources, as the hourly rates of external consultants are higher than that of the internal employees. The study by Bruce and Morris supports this by mentioning that in-house designers are less expensive to use compared with out-house designers (Bruce and Morris 1994). An additional consideration is that the practitioners participating in our study have various job responsibilities of, e.g. system developers, test managers and project managers. This means that they have to fulfil other tasks than just conducting usability evaluations, which means that when they spend time on conducting evaluations they cannot spend time on implementation and planning activities. These other tasks must then be completed at a different point in time.

7. Conclusions

The aim of the studies presented in this paper was to train SWPs from industry, which had no or minimum previous experience in usability work, to conduct usability evaluations. Based on this, we evaluated their performance compared to HCI specialists. We found that the practitioners were able to identify a considerable number of usability problems and that they performed on par with HCI specialists, which is explained by the higher level of domain knowledge.

Findings also showed that when acting in the role of test monitors, the practitioners facilitated good relations to the test users and they conducted a test in a structured manner following the time frame. They did, however, experience problems in making users think aloud and they also had a tendency to rescue the test users.

Additionally, practitioners were better at providing clear problem descriptions than at describing the impact, cause, user actions and data support. Their problem descriptions were of lower quality compared to an HCI specialist.

We furthermore found that the practitioners consistently gave lower severity ratings than the HCI specialists, which could be an effect of a subjective bias when evaluating own designs.

We also found that the practitioners committed to fixing most of the identified problems and that they also managed to eliminate most, which resembles the downstream utility found in other settings with established usability practices. These impact ratios indicate that the practitioners accepted and prioritised most of the problems, which deviates from the typical developer mindset found throughout the existing literature.

8. Limitations and future work

Due to the low number of participants ($n = 5$ in the initial study and $n = 3$ in the follow-up study) the level of generalisability becomes uncertain. The research methods applied

have warranted outcomes of qualitative and quantitative understandings of the barefoot evaluation approach. However, it may be argued that the outcome presented here is limited to the particular practitioners that participated and the case company only. On the other hand, the types of systems developed and perceived barriers to usability evaluations is similar to many companies from the same area in Denmark where this study has taken place (cf. Bak *et al.* 2008). Additionally, the study presented in Ardito *et al.* (2011) replicated the Danish study of Bak *et al.* (2008) within companies in southern Italy and had similar findings. Thus, it can be argued that our findings could be generalised to other companies with similar sizes, organisational structures and development methods. Nevertheless, it is relevant to conduct further studies with a higher number of participants and in particular companies with different profiles.

The systems evaluated in the initial and follow-up studies are web applications aimed for use by citizens and as such are work related. However, there are also a plethora of systems aimed for leisure where it would be more relevant to conduct evaluations of user experience rather than using classical usability metrics. For this reason, we cannot generalise our findings to evaluation to user experience. It can be argued though that practitioners could receive similar training to conduct evaluations using a different set of metrics.

Also, we focused exclusively on training practitioners to conduct usability evaluations. As Wixon (2003) points out, then it is equally important to tell the practitioners what to do and not just what is wrong within an interface. Thus, in the future it would be crucial to provide such practitioners with training in interaction design, which could decrease the number of recurring usability problems, i.e. this would further increase the impact of evaluations.

Finally, we find a need for studying sustainability of the barefoot approach imposed in the partnering company, which could be accomplished through longitudinal studies, i.e. do the practitioners conduct usability evaluations on their own in the long term? During the studies presented here, the authors were involved with the case company in the same research project. For this reason, we cannot entirely dismiss the presence of a Hawthorne effect. However, at the time of writing, a 20-month period has passed without any research activities in the case company. In that period, the practitioners have initiated three usability evaluations on their own, which indicates sustainability of the approach and, given that we did not participate in those evaluations, we can indeed dismiss the Hawthorne effect in that period.

Acknowledgements

We thank Mikael Skov, Kasper Hornbæk and Janne Jensen for the their work as external raters of problem descriptions as well as Kim Fiedler and Kaspar Lyngsie for their participation in analysing video data. We are also grateful to the SWPs and the case company who participated.

Funding

The research behind this paper was partly financed by the Danish Research Council [grant number 09-065143].

References

Ardito, C., *et al.*, 2006. Systematic evaluation of e-learning systems: an experimental validation. *In*: *Proceedings of NordiCHI*. New York: ACM Press, 195–202.

Ardito, C., *et al.*, 2011. Usability evaluation: a survey of software development organizations. *In*: *Proceedings of SEKE*. Knowledge Systems Institute Graduate School.

Bak, J.O., *et al.*, 2008. Obstacles to usability evaluation in practice: a survey of software development organizations. *In*: *Proceedings of NordiCHI*. New York: ACM Press, 23–32.

Bonnardel, N., Lanzone, L., and Sumner, T., 2003. Designing web sites: the cognitive processes of lay-designers. *Cognitive Science Quarterly*, 3 (1), 25–56.

Brown, C. and Pastel, R., 2009. Combining distinct graduate and undergraduate HCI courses: an experiential and interactive approach. *In*: *Proceedings of SIGCSE.* New York: ACM Press, 392–396.

Bruce, M. and Morris, B., 1994. Managing external design professionals in the product development process. *Technovation*, 14 (9), 585–599.

Bruun, A., 2010. Training software developers in usability engineering: a literature review. In: *Proceedings of NordiCHI*. New York: ACM Press, 82–91.

Bruun, A., 2011. Barefooted usability evaluation: addressing the mindset, resources and competences. *In*: *Proceedings of Interact*. Berlin: Springer, 374–377.

Capra, M.G., 2006. *Usability problem description and the evaluator effect in usability testing*. Thesis (PhD). Virginia Polytechnic Institute & State University.

Daqing, Z. and Unschuld, P.U., 2008. China's barefoot doctor: past, present, and future. *The Lancet*, 372 (9653), 1865–1867.

Edwards, A., Wright, P., and Petrie, H., 2006. HCI education: we are failing – why? *In*: *Proceedings of HCIEd*. Berlin: Springer, 127–129.

Følstad, A. and Hornbæk, K., 2010. Work-domain knowledge in usability evaluation: experiences with cooperative usability testing. *Systems and Software*, 83 (11), 2019–2030.

Fonseca, M., *et al.*, 2009. Conceptual design and prototyping to explore creativity. *IFIP*, 289, 203–217.

Frøkjær, E. and Hornbæk, K., 2008. Metaphors of human thinking for usability inspection and design. *TOCHI*, 14 (4), Article no. 20.

Frøkjær, E. and Lárusdòttir, M.K., 1999. Prediction of usability: comparing method combinations. *In*: *Proceedings of IRMA*. Idea Group.

Häkli, A., 2005. *Introducing UCD in a small-size software development organization*. Thesis (PhD). Helsinki University of Technology.

Hertzum, M., 2006. Problem prioritization in usability evaluation: from severity assessments toward impact on design. *Human-Computer Interaction*, 21 (2), 125–146.

Hertzum, M. and Jacobsen, N.E., 2001. The evaluator effect: a chilling fact about usability evaluation methods. *Human-Computer Interaction*, 13 (4), 421–443.

Høegh, R.T., *et al.*, 2006. The impact of usability reports and user test observations on developers' understanding of usability data: an exploratory study. *Journal of Human-Computer Interaction*, 21 (2), 173–196.

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

Howarth, J., Andre, T.S., and Hartson, R., 2007. A structured process for transforming usability data into usability information. *Journal of Usability Studies*, 3 (1), 7–23.

Juristo, N., Moreno, A.M., and Sanchez-Segura, M.I., 2007. Guidelines for eliciting usability functionalities. *IEEE Transactions on Software Engineering*, 33 (11), 744–758.

Kjeldskov, J., Skov, M.B., and Stage, J., 2004. Instant data analysis: conducting usability evaluations in a day. *In*: *Proceedings of NordiCHI.* New York: ACM Press, 233–240.

Koutsabasis, P., *et al.*, 2007. On the performance of novice evaluators in usability evaluations. *In*: *Proceedings of the Panhellenic conference on informatics*.

Law, E., 2006. Evaluating the downstream utility of user tests and examining the developer effect: a case study. *Human-Computer Interaction*, 21 (2), 147–172.

Medlock, M.C., 2002. Using the RITE method to improve products: a definition and a case study. *In*: *Proceedings of UPA*. UPA.

Molich, R., 2000. *User-friendly web design*. Copenhagen: Ingeniøren Books.

Nielsen, J., 1992. Finding usability problems through heuristic evaluation. *In*: *Proceedings of CHI*. New York: ACM Press, 373–380.

Nielsen, J., 1993. Iterative user-interface design. *Computer*, 26 (11), 32–41

Nielsen, J., 1994. Usability inspection methods. *In*: *Proceedings of CHI*. New York: ACM Press, 413–414.

Rosenbaum, S., Rohn, J.A., and Humburg, J., 2000. A toolkit for strategic usability: results from workshops, panels, and surveys. *In*: *Proceedings of CHI.* New York: ACM Press, 337–344.

Rubin, J. and Chisnell, D., 2008. *Handbook of usability testing: how to plan, design, and conduct effective tests*. 2nd ed. Indianapolis, IN: John Wiley.

Sawyer, P., Flanders, A., and Wixon, D., 1996. Making a difference – the impact of inspections. *In*: *Proceedings of CHI.* New York: ACM Press, 376–382.

Schaffer, E., 2007. *Institutionalization of usability: a step-by-step guide*. Redwood City, CA: Addison-Wesley.

Scholtz, J., Laskowski, S., and Downey, L., 1998. Developing usability tools and techniques for designing and testing web sites. *In*: *Proceedings conference on human-factors & the web*.

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* New York: ACM Press, 1–9.

Skov, M.B. and Stage, J., 2005. Supporting problem identification in usability evaluations. *In*: *Proceedings of OzCHI*. Computer-Human Interaction Special Interest Group (CHISIG) of Australia. New York: ACM Press, 1–9.

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.

Skov, M.B. and Stage, J., 2009. Training software developers and designers to conduct usability evaluations. *Behaviour & Information Technology*, 31 (4), 425–435.

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

Wixon, D., 2003. Evaluating usability methods: why the current literature fails the practitioner. *Interactions*, 10 (4), 28–34.

Wright, P.C. and Monk, A.F., 1991. The use of think-aloud evaluation methods in design. *ACM SIGCHI Bulletin Archive*, 23 (1), 55–57.