The Journal of Systems and Software 105 (2015) 40–53

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

New approaches to usability evaluation in software development: Barefoot and crowdsourcing

Anders Bruun*, Jan Stage[1]

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

ARTICLE INFO

Article history:
Received 12 May 2014
Revised 14 March 2015
Accepted 17 March 2015
Available online 21 March 2015

Keywords:
Usability
Developers
Users

ABSTRACT

Usability evaluations provide software development teams with insights on the degree to which software applications enable users to achieve their goals, how fast these goals can be achieved, how easy an application is to learn and how satisfactory it is in use. Although such evaluations are crucial in the process of developing software systems with a high level of usability, their use is still limited in small and medium-sized software development companies. Many of these companies are e.g. unable to allocate the resources that are needed to conduct a full-fledged usability evaluation in accordance with a conventional approach.

This paper presents and assesses two new approaches to overcome usability evaluation obstacles: a barefoot approach where software development practitioners are trained to drive usability evaluations; and a crowdsourcing approach where end users are given minimalist training to enable them to drive usability evaluations. We have evaluated how these approaches can reduce obstacles related to limited understanding, resistance and resource constraints. We found that these methods are complementary and highly relevant for software companies experiencing these obstacles. The barefoot approach is particularly suitable for reducing obstacles related to limited understanding and resistance while the crowdsourcing approach is cost-effective.

1. Introduction

Usability is a quality attribute of a software application that reflects "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (ISO, 1998). Usability evaluations provide software development teams with insights on the degree to which software application enable users to achieve their goals, how fast these goals can be achieved, how easy an application is to learn and how satisfactory it is in use (Rubin and Chisnell, 2008).

Despite the general agreement that usability evaluations are crucial in the process of developing software systems with a high level of usability, their adoption is still limited in small and medium-sized software development companies (Bak et al., 2008; Ardito et al., 2011).

1.1. Obstacles for adopting usability practices

Studies from around year 2000 started to examine a range of obstacles prohibiting adoption of usability practices in software development companies. The obstacles identified in some of the first studies by Gunther et al. (2001) and Rosenbaum et al. (2000) include resistance towards usability practices by members of development teams, limited understanding of the usability concept and resource constraints. A few years later, Gulliksen et al. (2004) made a similar study uncovering factors for successful adoption of usability practices in Swedish companies. One of the key factors was to obtain acceptance from the development team, which is related to the resistance obstacle identified in the earlier studies. More recently, Bak et al. (2008) and Ardito et al. (2011) studied obstacles in Danish and Italian companies, respectively. Bak et al. (2008) found the main obstacles to be perceived resource constraints, limited understanding of the usability concept and resistance among development team members towards adopting usability practices. Similarly, Ardito et al. (2011) found the main obstacle to be related to perceived resource constraints. These studies have identified several causes for limited adoption of usability practices, but they generally agree that the three main obstacles are perceived resource constraints, limited understanding of the usability concept and methods, and developer resistance towards adopting usability practices.

Even though these obstacles have been known for over a decade, they are still highly relevant. Perceived resource constraints are especially important in small and medium-sized software development companies. Typically, such companies do not have funding for comprehensive consultancy or hiring usability specialists (Häkli, 2005; Juristo et al., 2007; Scholtz and Downey, 1998) as they are exceedingly expensive (Nielsen, 1994). The resistance obstacle concerns the level

* Corresponding author. Tel.: +45 9940 9971; fax: +45 9940 9798.
E-mail addresses: bruun@cs.aau.dk (A. Bruun), jans@cs.aau.dk (J. Stage).
[1] Tel.: +45 9940 8914.

http://dx.doi.org/10.1016/j.jss.2015.03.043
0164-1212/

of acceptance, where problems identified through usability evaluations are not always accepted by members in the development team (Bak et al., 2008). Resistance also encapsulates low priority on fixing of identified usability problems whereas implementation of functionality and bug fixing have higher priority (Bak et al., 2008). Limited understanding of usability reflects that software development practitioners and management have no (or very limited) knowledge of the usability concept and the related core methods (Gulliksen et al., 2004).

1.2. Overcoming the obstacles

The literature on usability evaluation includes a variety of means that have been proposed to overcome the obstacles towards adoption of usability practices in software development companies. A significant number of these proposals are based on the idea that software development practitioners should conduct their own usability evaluations. It is typically argued that if software development practitioners are enabled to conduct usability evaluations, it will lessen the need in small and medium-sized companies to employ usability specialists, and this will resolve challenges in relation to resource constraints. Moreover, letting software development practitioners conduct usability evaluations will provide them with first-hand observations of users, which in turn will overcome the obstacles related to limited understanding and resistance. Software development practitioners are generally lacking usability evaluation skills (Gulliksen et al., 2004), thus some proposals employ this approach:

(1) A method approach where software development practitioners are provided with methods to support them in conducting usability evaluations

The assumption is that software development practitioners can conduct their own usability evaluations if they are provided with the right methods. An early example of this is Nielsen's (1992) study of the performance of specialists, non-specialists and double-experts in conducting heuristic inspection.

A different group of proposed solutions to overcome the key obstacles focus on tools instead of methods, thereby relating to this approach:

(2) A tool approach where software development practitioners are provided with tools to support them in conducting usability evaluations

The assumption is that a tool can replace some of the skills that are needed when conducting a usability evaluation. The tool can be either a software tools or a conceptual tool. One example is a software tool aimed to support the transformation of raw usability data into usability problem descriptions (Howarth, 2007; Howarth et al., 2007). Another example is a conceptual tool aimed to support identification of usability problems in a video recording or a live user session (Skov and Stage, 2005).

The method and tool approaches have been studied to a great extent. Less research has been committed to the idea of training software development practitioners who are usability novices to conduct usability evaluations. This is summarized as:

(3) A barefoot approach where software development practitioners are trained to drive usability evaluations

The assumption is that software development practitioners who have little to no knowledge on how to conduct usability evaluations can be trained to achieve basic evaluation skills. Häkli (2005) presents a study in which she trained software development practitioners without a usability background to conduct heuristic inspections and user based evaluations. Høegh et al. (2006) conducted a study of usability evaluation feedback formats where they examined how software development practitioners' awareness of usability problems could be increased; one of these formats was to let practitioners observe a user-based evaluation and thereby involve them directly in the process. Although no training was done in that study, it is an example of including the developers in the evaluation process in order to increase awareness.

An entirely different idea is to involve end-users in usability evaluations. It is argued that if end users are able to conduct such evaluations, it will lessen the need for small companies to employ usability specialists. This can be summarized in this approach:

(4) A crowdsourcing approach where end users are given minimalist training to enable them to drive usability evaluations

The assumption is that end users provided with minimalist training in driving usability evaluations will alleviate the need to involve usability specialists. This was originally proposed by Castillo et al. (1998) as a feasible alternative to traditional usability evaluations conducted by usability specialists. A main purpose of their User reported Critical Incident (UCI) method was to reduce the amount of resources required for having usability experts analyze data from system usage. Instead, users would receive minimalist training in identifying and describing usability problems after which they would report the problems (Castillo et al., 1998).

1.3. The barefoot approach

The barefoot approach has been suggested as a way of overcoming the key obstacles towards adoption of usability evaluation practices. With this approach, existing software development practitioners are trained to plan and conduct usability evaluations and to take on the data logger and test moderator roles. The practitioners are also trained to analyze usability data and fix identified problems. These developers would continue doing their usual development tasks, but would also be conducting usability evaluations.

This approach inherits the idea behind the barefoot doctors that emerged during the Cultural Revolution in China in the 1960s. According to Daqing and Unschuld (2008), getting healthcare services embedded in the rural areas of China was an ongoing challenge dating back to the early 20th century. Early attempts to resolve this challenge included drafting doctors from private practices, but healthcare services in these areas remained scarce. In 1964, the Chinese state covered healthcare expenditures for 8.3 million urban citizens, which exceeded the expenditures for more than 500 million peasants residing in rural areas. Mao Zedong criticized this urban bias of healthcare services, and in 1965 he emphasized the importance of solving this challenge. Accordingly, one vision behind the Cultural Revolution was to bring better healthcare services to the rural areas. 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 delivery of babies, ensuring better sanitation and performing simple surgical procedures. In order to keep up the level of mass production, peasants who received this basic 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 (Daqing and Unschuld, 2008).

Although barefoot doctors did not have the same level of competences and equipment as urban doctors, the barefoot programme did, according to the World Health Organization (WHO), effectively reduce healthcare costs as well as provide timely care. Thus, the WHO considered the barefoot doctors programme successful in terms of solving the challenge of healthcare shortages (Daqing and Unschuld, 2008).

1.4. The crowdsourcing approach

The crowdsourcing approach means that users drive usability evaluations in the sense that they solve a set of predefined tasks and report the problems they experience in a remote asynchronous setting. The concept behind remote asynchronous methods is that the end-user and usability evaluator are separated in space and time, i.e. the user evaluates a system and provides comments without the physical presence of an evaluator and at a time that suits them (Castillo et al., 1998). This is similar to the idea of crowdsourcing; Doan et al. (2011) describe this by illustrating various systems, e.g. they identify types of crowdsourcing systems for the web and identify benefits as well as challenges in relation to these. The overall purpose of crowdsourcing systems is to recruit numerous users to help solve different types of problems as is the case of Wikipedia and systems for open source software development. A crowdsourcing system is defined as a system that "... enlists a crowd of humans to help solve a problem defined by the system owners" (Doan et al., 2011).

Various crowdsourcing systems exist and they involve different kinds of user contributions. Some crowdsourcing systems enable users to evaluate artifacts such as books or movies and others let users share content in the form of products or knowledge. An example of sharing a product can be found in peer-to-peer services where users share music files and Wikipedia is a classic example of users sharing knowledge. Other crowdsourcing systems enable social networking between users while others depend on users building artifacts, e.g. open source software or execute tasks (Doan et al., 2011).

There are four challenges to crowdsourcing that have been emphasized (Doan et al., 2011). The first challenge is how users can be recruited and retained. There are four solutions to this challenge: requiring users to participate, paying users, asking users to volunteer and making users pay for a service. Given the vast amount of users on the web, the solution of asking for volunteers is mentioned as being free and easily executed, which makes this the most popular approach (Doan et al., 2011). The second challenge is related to the types of contributions that users can make. In principle, any non-trivial problem can benefit from crowdsourcing. However, it is important to consider how cognitively demanding the contributions are compared to the types of users providing them. The third challenge is combining contributions, which is a relatively simple task when users provide quantitative data, e.g. numeric ratings, as this can be done automatically. However, when users provide qualitative contributions such as free form texts, a higher degree of manual labor is required. Finally, the fourth challenge is that of evaluating users and their contributions. As crowdsourcing is based on contributions from several users, of which some may be naive, there is a need to filter the data. One solution to this can be to delimit certain types of users of making complex contributions and other solutions include manual or automatic detection of users providing malicious content, e.g. where the system asks questions to which answers are already known (Doan et al., 2011).

The idea of remote asynchronous usability evaluation is similar to the idea of crowdsourcing as a group of users are enlisted in order to solve a problem. The problem to be solved in this case is the identification of usability deficiencies where users are recruited to evaluate an artifact in the form of a user interface. This fits well to the application of the UCI method (Castillo et al., 1998).

1.5. Objective

In this paper we discuss to what extent the barefoot and crowdsourcing approaches to usability evaluations can help software development companies overcome the three critical obstacles related to resource constraints: limited understanding of the usability concept and methods, the cost of conducting usability evaluations and the developer resistance towards adopting usability practices. The discussion is based on four case studies that have previously been published individually. This paper extends the individual studies by providing an overview of each of the two approaches, including the fundamental theoretical considerations behind it. In addition, we discuss the relative strengths and weaknesses of these two approaches in overcoming the three main obstacles. We also emphasize the complementarity of the two approaches and provide guidance to software development companies in selecting the approach that maximizes their outcome and thereby lower the barrier for them to engage in usability evaluations.

2. Related work

The objective of this paper is to assess the feasibility of barefoot usability evaluations conducted by development practitioners as well as crowdsourcing evaluations conducted by end users. We have found no studies encompassing both approaches, which is why we in this section provide two separate overviews of related work. The first subsection presents studies related to the barefoot approach. This is followed by an overview of related work regarding usability tests based on crowdsourcing.

2.1. Training software development practitioners

This section provides an overview of a previously conducted literature survey on training novices in usability engineering methods, cf. Bruun (2010). Here we provide a condensed description of empirically based studies of novice usability evaluators conducting user based usability tests. Based on the literature survey we identified three research foci: studies of tools, studies of methods and studies of training.

Howarth et al. (2007) and Skov and Stage (2005) study software tools or conceptual tools developed to assist usability evaluators in identifying problems. The study presented in Howarth et al. (2007) aims to develop and evaluate a software tool, which eases transformation of raw usability data into usability problem descriptions. In that experiment, 16 graduate students acted as usability evaluators by applying one of two software tools to describe problems. They received an hour of training to get to know the tools and were then asked to view videos from a previously conducted usability test. Their performance was measured in terms of problem description quality based. Findings showed that students were better at formulating user actions than providing clarity, data support etc. in their problem descriptions. In Skov and Stage (2005) a study on developing and evaluating a conceptual tool is presented. That tool is a 4×3 matrix that evaluators can apply when observing users in order to facilitate problem identification and categorization. Fourteen undergraduate students participated in a comparative study consisting of two experimental conditions; one condition in which the tool was applied to test a user interface and another without the tool. Students viewed recordings from a previous usability test and findings in that study primarily regarded thoroughness. In that paper it is shown that students identified 18% of all problems and discovered a mean of 20% of the problems identified by two usability specialists.

Other papers emphasize comparative studies of usability testing methods. The study presented in Koutsabasis et al. (2007) describes an experiment on evaluating the performance of students in terms of thoroughness and reliability. That study applies 27 graduate students as the empirical basis, and they were distributed over four conditions: heuristic Inspection, Cognitive Walkthrough, user based testing 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) and that reliability was 11% on average. It should be noted, that reliability in that study is based on problem agreement between methods and not internal agreement between evaluators applying a specific method. In Ardito et al. (2006)

the development and evaluation of the eLSE method is described. Here 73 senior students participated in comparing the performance of several evaluation methods; eLSE, user based testing and Heuristic Inspection. Findings in that study show that students applying the user based testing had an average thoroughness of 11%. Frøkjær and Lárusdòttir (1999) present a comparative study of usability testing methods which emphasizes the effect of combining methods. Their study is based on 51 students who in one experimental condition applied Cognitive Walkthrough followed by a second round of a user based testing. In a second condition other participants applied Heuristic Inspection followed by user based testing. Findings in that study reveal a thoroughness of 18%.

Finally, some studies emphasize training of novices in analyzing data from user based tests. As an example, the students participating in the study presented in Skov and Stage (2009) received 40 hours of training beforehand. They were then instructed to conduct a usability test, analyze the results and to write a report documenting all steps in the process. Student reports were then compared to reports made by usability specialists. Results show that the students revealed a mean 37% of the problems identified by specialists. Wright and Monk (1991) also emphasize training aspects. Their paper presents two experiments where both aim to study the application of user based tests. In the first experiment software trainees read a short manual after which they were asked to conduct a user based usability test. As in Skov and Stage (2009), trainees documented identified problems in a report and these were assessed in terms of thoroughness and problem severity. The second experiment of that study aims to examine differences in testing your own design versus the design made by others. In one experimental condition trainees designed and tested their own design and in the second they tested designs made by others. Reports were also assessed with respect to thoroughness. Findings revealed that all student teams identified 33% of all problems on average. Häkli (2005) is a master thesis describing the process of introducing a user-centered method in a small software company through training. The researcher conducted a 14 hour training course for software development practitioners. The topics of the course were interaction design, prototyping, Heuristic Inspection and user based testing. Emphasis in that study is on participant performance in conducting Heuristic Inspections. Few qualitative observations on e.g. how well the practitioners performed as test monitors in a user based test were reported. One finding is 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.

2.2. Training end users

In the literature we have identified little over 20 papers presenting empirically based studies of remote asynchronous methods, see Bruun et al. (2009) and Bruun and Stage (2012a,b) for further details. Through the literature survey, we found that remote asynchronous tests mainly have been applied for conducting summative evaluations and fewer report using this method for formative purposes.

Auto logging is applied to collect quantitative data such as URL history and task completion times, i.e. this remote asynchronous method indicates if the paths to complete tasks are well designed (Scholtz, 1999; Scholtz and Downey, 1998; Steves et al., 2001; Waterson et al., 2002; West and Lehman, 2006; Winckler et al., 1999, 2000). The method, however, lacks the ability to collect qualitative data, which is needed to address usability problems beyond the likes of path finding and time used. This makes auto logging particularly well suited for conducting summative tests. For this reason, auto logging is often supplemented by other data collection methods such as interviews and questionnaires. In combination, auto logging and interviews/surveys found many of the same problems as heuristic inspection (Steves et al., 2001). In Scholtz and Downey (1998) evaluators applied this approach and revealed 60% of the problems found via a heuristic inspection. In Winckler et al. (2000) it was found that auto logging is not as efficient as a conventional laboratory method in terms of thoroughness. The studies presented in Steves et al. (2001) and Winckler et al. (2000) do not provide details about the thoroughness of auto logging. In another study evaluators applying the auto logging method identified 40% of the usability problems found in a conventional laboratory test (Waterson et al., 2002). An online discussion forum has also been proposed as a source for collecting usability feedback (Millen, 1999). In that study a forum was used as a source for collecting qualitative data in combination with auto logging. Participants were in that case not explicitly asked to report usability problems through the forum, but the participants did still provide detailed usability feedback. Millen (1999) provides no information about training or the number of usability problems found using this approach. Thompson (1999) argues that participants may be further motivated to report usability issues when making reporting a collaborative effort, which is supported through the forum, but we have found no empirical studies to support this claim. Finally, Steves et al. (2001) studied the approach of supplementing auto logging with a diary where participants provided qualitative information on the problems identified. We have found no information about the usefulness of this approach and the experiences with it. Steves et al. (2001) mention that participants used their diary on a longitudinal basis to report on the usability problems they experienced with the use of a hardware product.

The remote asynchronous method of User reported Critical Incident (UCI) is suitable for conducting formative evaluations as it provides insights as to why users experience particular problems. UCI is based on the idea that the users themselves report the problems they experience in a structured web form, which relieves evaluators from conducting the evaluation and analyzing results (Castillo, 1997). The User-reported Critical Incident method (UCI) is based on the idea that the users themselves report the problems they experience. This should relieve evaluators from conducting tests and analyzing results, but it requires training as the end users need to know how to identify and describe usability problems (Castillo et al., 1998). It has been concluded that test participants are able to report their own critical incidents, e.g. Castillo shows that a minimalist approach works well for training participants in identifying critical incidents (Castillo, 1997). The first studies of this method concluded that the participants were able to categorize the incidents (Castillo, 1997; Castillo et al., 1998), but a more recent and systematic study concludes that the participants cannot categorize the severity of the critical incidents they identify (Andreasen et al., 2007). The reason for this discrepancy may be that the training conducted by Castillo and colleagues was more elaborate than the training provided by Andreasen et al. (2007). However, in Castillo (1997) and Castillo et al. (1998) the training was done with the researchers physically present which contradicts the idea of remote testing. The training conducted by Andreasen et al. (2007) was done over the internet. The number of usability problems identified also varies between the different studies. In one of the first studies, 24 participants identified 76% of the usability problems found by experts (Castillo et al., 1998). In a later study, 10 participants identified 60% of the problems found in a conventional laboratory test (Thompson, 1999). In Andreasen et al. (2007) 6 participants identified 37% of the usability problems found in a conventional laboratory test.

3. Case studies

Over a period of three years we conducted two case studies to validate the barefoot approach and another two case studies to validate the crowdsourcing approach. Findings from these four studies are bound together in this paper. The studies of the barefoot approach can be found in Bruun and Stage (2011) and Bruun and Stage

Table 1
Overview of the software development practitioners' (SWP) 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

(2012b) and the studies of the crowdsourcing approach can be found in Bruun et al. (2009) and Bruun and Stage (2012a). In this section we provide a general overview of the four case studies. For further details we refer to the above mentioned papers.

Hornbæk derived a list of "Dogmas in the assessment of usability evaluation methods" (Hornbæk, 2010). Referring to the first dogma in Hornbæk (2010), it was critical for us to not only evaluate the number of problems identified, but also include other metrics relevant for practice, e.g. the extent to which identified problems are fixed and the cost-effectiveness of the methods.

3.1. Barefoot studies

In these studies software development practitioners conducted a total of three usability tests of which the first was conducted in the laboratory at the university. Our aim of this first test was to evaluate the ability of the practitioners to act as test monitors and to assess the feasibility of the barefoot approach in terms of thoroughness, i.e. their ability to detect usability problems, cf. Bruun and Stage (2011). A follow-up study was conducted where the practitioners from the same company conducted two usability tests. The aim of the follow-up study was to evaluate performance of the barefoot approach regarding downstream utility, cf. Bruun and Stage (2012b).

3.1.1. Participants

Eight software development practitioners (SWPs) participated in the feasibility and follow-up studies. All were employed in the same small software development company with 20 employees. The company develops web applications for the public sector. The subsection "System" below provides an example of the applications developed. Table 1 shows 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 software development practitioners had previous practical experience of conducting usability evaluations where SWP 1 as part of his education attended an HCI course and had experience in the conduction of 4–5 usability evaluations 7 years prior to our studies. SWP 5 had also attended an HCI course during his education and had experience in conducting a single usability evaluation 13 years prior to these studies. 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 2 years prior, 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 knowledge of usability evaluations, we argue that the SWPs participating in our experiments were not HCI specialists.

3.1.2. Training course

Since the practitioners had no or limited experiences in conducting usability tests we focused on teaching a traditional user based test with video analysis as described in Rubin and Chisnell (2008). This was held as a 2-day (14 h) training course by the authors of this paper. Teaching was done as a combination of presentations and exercises and to conclude this part of the training the practitioners were asked to do a homework assignment of analyzing five video clips from a previous usability test. The resulting problem lists were reviewed and we provided feedback to the practitioners on how to improve their problem descriptions.

This traditional usability test necessitates traversing several hours of video data, which require a considerable amount of resources. Therefore we also chose to train the practitioners in applying Instant Data Analysis (IDA). IDA is not based on reviewing video data and is conducted immediately at the end of a test and involves the following steps, cf. (Kjeldskov et al., 2004):

- 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.
- Task review: the test monitor and data loggers review all tasks to recall additional problems that occurred.
- Note review: the data loggers review their notes to remember further problems.
- Severity rating: the test monitor and data loggers discuss the severity of the problems and rate these as critical, serious or cosmetic, cf. Molich (2000).

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

3.1.3. Conduction of the usability tests

The SWPs planned all usability tests by making a test plan according to Rubin and Chisnell (2008), which included reflections on test type (formative/summative), types of users, location, equipment, test tasks and roles between the SWPs. The SWPs also identified and invited representative end users to participate in the tests without the involvement of the researchers.

3.1.3.1. System. Two different systems were evaluated during the two case studies. In the first case study this was a web application that citizens use when moving from one address to another. The Danish municipalities 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 practitioners were employed.

In the follow-up study, the practitioners evaluated 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. This system was also developed by the small software company in which the practitioners were employed.

3.1.3.2. Setting. Fig. 1 shows the setting applied in the first evaluation conducted at the university lab (feasibility study) and Fig. 2 shows the setting applied in the two final evaluations of the follow-up study. Tests in the follow-up study were situated in an office at the case company. In all tests we configured video capture software, audio recording equipment and a camera capturing the facial expressions of the users. Fig. 3 illustrates the picture-in-picture recordings made in all tests.

During each test session a user was sitting at a table solving the predefined tasks by using the system. One SWP acted as test monitor and sat next to the users. Other SWPs acted as data loggers and noted down usability problems. A total of 13 representative end users participated in the evaluations: six in the usability test of the feasibility study and seven in the two evaluations of the follow-up study.

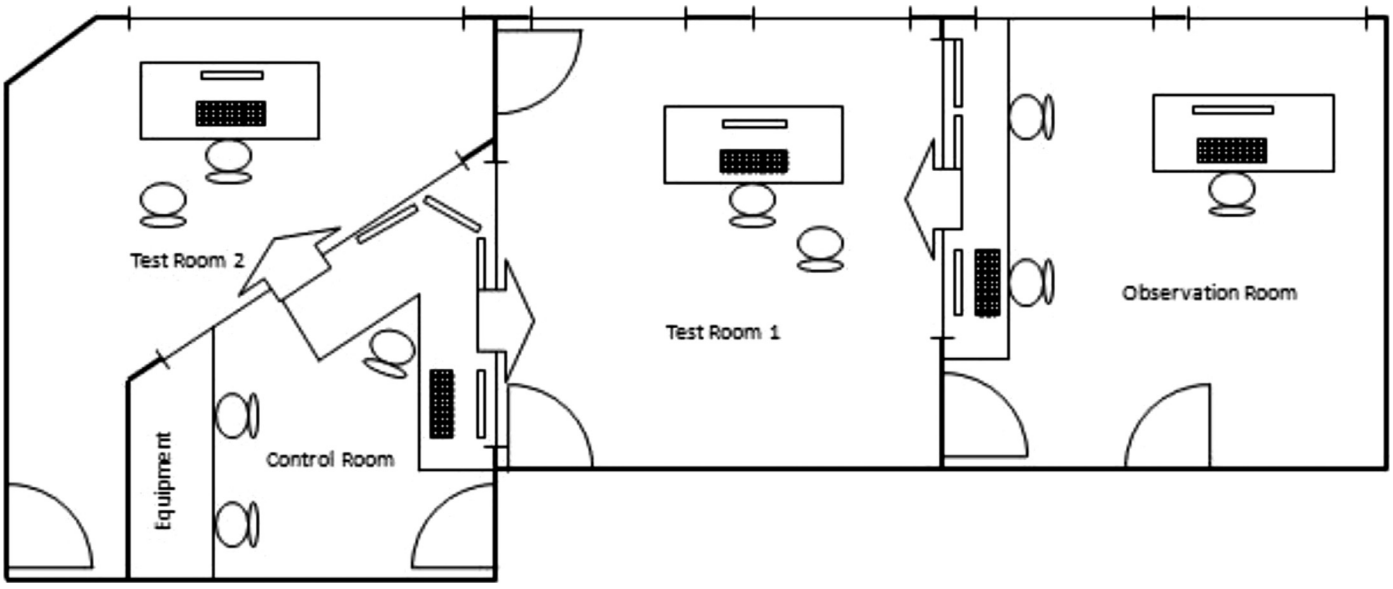

Fig. 1. Layout of the usability laboratory (first usability test).

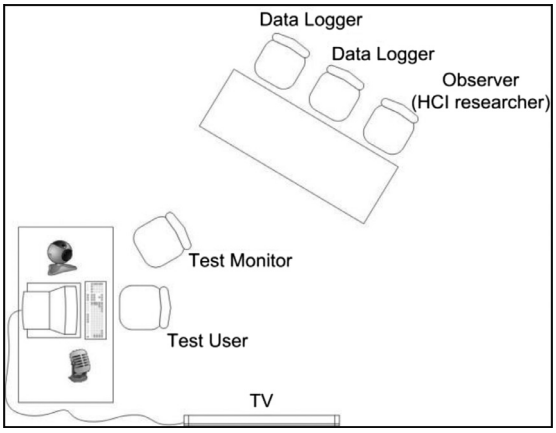

Fig. 2. Layout of the office in the case company (final two usability tests).

3.1.3.3. Procedure. The usability test in the feasibility study was planned and conducted by SWPs 1, 2 and 3 (Table 1), who also took turns in acting as test monitors. These, plus SWPs 4 and 5 analyzed the obtained video material and described the identified usability problems. The system for changing addresses was evaluated.

The two usability tests in the follow-up study were planned and executed by SWPs 6, 7 and 8. The initial version of the wage subsidy system was evaluated in the first test. They applied Instant Data Analysis as an alternative to conventional video based analysis (which was applied in the feasibility study). These three practitioners also took turns in acting as test monitors. SWPs 6, 7 and 8 were responsible for the development of the wage subsidy system and the matched list of usability problems was used as input to improve the usability in a new version. Two days after the first test they held a one-day meeting discussing what problems to fix and 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. Three months later, the updated version of the system was evaluated by the same SWPs.

3.1.3.4. Data analysis . For each usability test, the SWPs analyzed data individually and held meetings afterwards in which they matched their individual findings into a complete list of problems. Following suggestions made in Hornbæk (2010), all evaluators applied the same format for describing usability problems as this should increase reliability of the matching procedure (described below). The feasibility study was based on SWPs conducting classical video based analysis and they extracted problems using the framework suggested in Skov and Stage (2005). The two tests in the follow-up study were based on SWPs conducting Instant Data Analysis where problems were extracted using the three-step process proposed in Kjeldskov et al. (2004).

For comparison purposes we additionally had three usability specialists review the video recordings from all tests. Two of the specialists had not otherwise taken part in planning or conduction of the experiment, i.e. these are considered to be unbiased. The three specialists applied the same procedure for video analysis as the SWPs in the feasibility study.

Finally, the HCI specialists held meetings with the SWPs in order to match usability problems into a white list of problems for each of the tests. These white lists enabled us to evaluate the number of problems identified by SWPs and specialists and it served as a basis for assessing downstream utility. All evaluators were instructed to apply the same structured format for describing usability problems. This was done to ensure that problems were described at similar levels of granularity. Additionally, we did conduct problem matching in teams to counteract the evaluator effect as suggested in Hornbæk and Frøkjær (2008).

3.2. Crowdsourcing studies

We conducted two case studies where we provided minimalist training of end users to conduct usability tests and to describe the problems experienced. The purpose of the first study was to evaluate the feasibility and cost effectiveness in identifying problems through the User reported Critical Incident method, cf. Bruun et al. (2009). The aim of the second study was to evaluate effectiveness of different types of user training, cf. (Bruun and Stage, 2012a).This is critical as it is complicated to teach users to conduct remote usability evaluations, mainly because training has to be done remotely in order to harvest the full potential of the method.

3.2.1. Participants

In the first crowdsourcing study we recruited 20 university students of which half were female. Half of the participants were

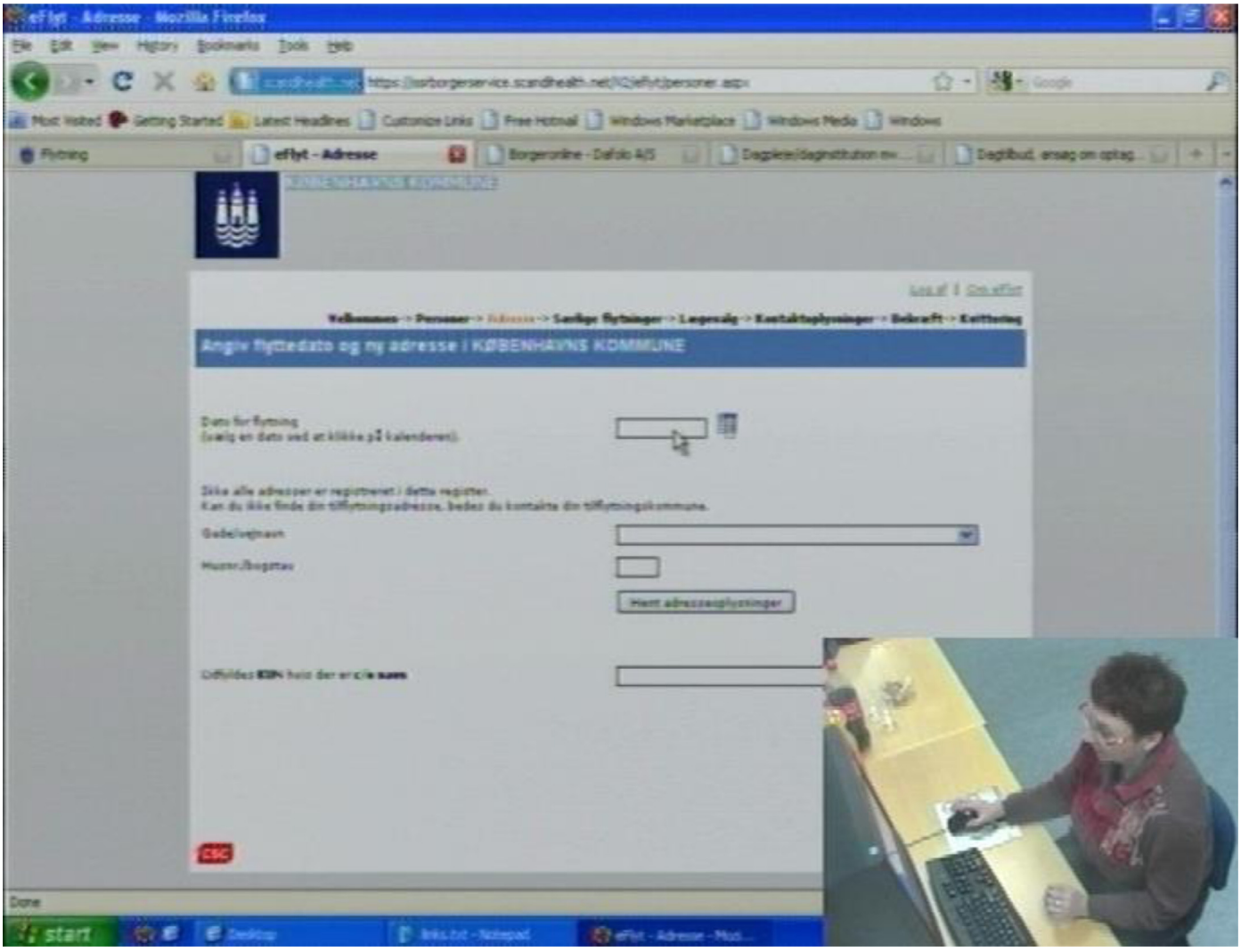

Fig. 3. Snapshot of video recording.

undertaking a technical education e.g. computer science or electrical engineering and the other half a non-technical e.g. communication and social science. In the second crowdsourcing study we recruited 33 university students undertaking an education within the school of ICT. Each student only participated in one of the studies.

3.2.2. Minimalist training

Participants in both crowdsourcing studies received minimalist training on how to identify and describe usability problems. Training material was conveyed in an online written form consisting of a combination of deductive and inductive instructions of approximately one A4 page of length. Deductive instructions is the typical way of conveying information in e.g. engineering and science (Prince and Felder, 2006) and denotes the pedagogical approach where the teacher presents a general rule or definition to be learned, after which the learners reason on observations or examples that fit within the rule. In this case learners are told up front exactly what they need to know, which makes it a straightforward and well-structured approach to teaching. As an alternative pedagogical approach there are inductive instructions. Here specific observations or examples are presented initially, and then learners infer the general rule (Prince and Felder, 2006). The examples must be familiar to the learners in order to create the best possible conditions for them to assimilate the new knowledge within their existing knowledge structures (Prince and Felder, 2006). Some learners are best stimulated by deductive instructions while others prefer induction (Felder and Silverman, 1988), which is why we based our instructions on a combination of deduction and induction. In practice we did this by providing a definition of how a usability problem is defined (deduction) combined with two examples of usability problems experienced using Facebook and Microsoft Word (induction).

3.2.3. Conduction of the usability tests

3.2.3.1. System. The system evaluated in the first crowdsourcing study was the open source email client Mozilla Thunderbird. In the second study participants were asked to evaluate the website of the School of ICT, which provides information on study regulations, exam schedules, campus maps etc.

3.2.3.2. Setting. We did not impose any requirements on the participants regarding the setting, i.e. participants in both studies worked at home or at the university using their own computers at a time suitable for them. We did, however, ask them to participate within a timeframe of 3–4 weeks. In the first study we had 10 participants applying the UCI method and 23 in the second.

In both crowdsourcing studies we also conducted a conventional laboratory evaluation for comparison purposes. In both studies we had 10 participants in the lab condition. The laboratory tests were conducted in lab settings similar to that applied in the barefoot studies

illustrated in Fig. 1. Here we also recorded video data in a picture-in-picture setup.

3.2.3.3. Procedure. All participants received training material and task instructions via email. They were then asked to firstly walk through the training material and then start to solve the predefined tasks that we provided for the systems. In the first study we provided nine tasks relevant for evaluating an email client, e.g. create a mail account, edit contact info add a spam filter etc. In the second study we also provided nine tasks related to the use of the School of ICT website, e.g. finding study regulations, exam dates, contact persons etc.

In both studies we instructed participants to report any usability problems identified using the systems as soon as they discovered these. This is in line with the User reported Critical Incident (UCI) method proposed by Castillo et al. (1998). Participants were reporting problems through a web form developed using PHP, JavaScript and a MySQL database. All participants received a unique login and a link to the online report form, which was similar to the form used in other UCI experiments (Castillo, 1997; Castillo et al., 1998).

The following points had to be answered using this form:

- What task were you doing when the critical incident occurred?
- What is the name of the window in which the critical incident occurred?
- Explain what you were trying to do when the critical incident occurred.
- Describe what you expected the system to do just before the critical incident occurred.
- In as much detail as possible, describe the critical incident that occurred and why you think it happened.
- Describe what you did to get out of the critical incident.
- Were you able to recover from the critical incident?
- Are you able to reproduce the critical incident and make it happen again?
- Indicate in your opinion the severity of this critical incident.

At the end of the web form page there was a submit button and when pressed, the data were saved in the MySQL database and the form was reset, ready for a new entry. The form was active in a separate browser window, so the participants could switch between the system under evaluation and the web form each time they encountered a problem.

3.2.3.4. Data analysis . In both studies all data was collected before conducting the analysis. In the first study we had 10 problem reports from UCI and 10 user videos. In the second study we had 23 problem reports from UCI and 10 user videos. Similar to the barefoot studies, we had external (unbiased) evaluators review video recordings from the laboratory tests. In the first crowdsourcing study three external evaluators, who had not taken part in the design and conduction of the experiments, analyzed the video material. In the second study we had one of the authors of this paper and two external evaluators analyzing the video data. In case of the lab condition, the videos were thoroughly analyzed trough a classical video analysis, cf. (Rubin and Chisnell, 2008). Usability problems from these videos were extracted on the basis of the framework in Skov and Stage (2005).

The data sets from the UCI conditions in both studies were analyzed by reading one problem report at a time. These datasets were analyzed by the authors of this paper and four external evaluators. By using only the information available in the users' reports, it was transformed into a usability problem description. If necessary, the email client or website was checked to get a better understanding of the problem. Evaluators individually analyzed all the data sets one at a time in random order.

Problem reports from the UCI conditions in both experiments were validated by considering the comprehensiveness of the wording, i.e. that the problem was formulated in such a way that we could understand the problem and locate it in the user interface. This is similar to the validation procedure described in Bosenick et al. (2007). If a description could not be translated into a meaningful problem in short time or the problem could not be identified using the website, the problem was not included in the problem list. Validity of observations, i.e. whether a detected problem is "real" or not has been discussed at length within usability evaluation literature in relation to non-user based inspection methods. Hartson et al. (2001) introduced the notion of a "real" usability problem and defined such a problem to be real if: "*... it is a predictor of a problem that users will encounter in real work-context usage and that will have an impact on usability ...*". This definition relies on the assumption that usability evaluations should be conducted with end users, and Hartson et al. also note that non-user based inspections reveal false positives. Since the UCI method is user based we consider identified problems to be valid. Furthermore, Molich and Dumas (2008) revised the discussion of false positives in a recent study. They found no clear evidence of false positives identified through inspections compared to conventional user based evaluation (Molich and Dumas, 2008).

After completing the individual analysis the evaluators met in order to match the problem lists such that we had one total list of problems for each laboratory and UCI evaluation. This is similar to the approach taken in the barefoot studies described previously. Furthermore, to increase reliability of the matching process, the evaluators conducting video analysis described usability problems using the same structured format. This was also the case for the participants using UCI to report problems remotely.

4. Results

In this section we present findings from our barefoot and crowdsourcing studies. We present findings in relation to the number of problems detected, downstream utility and cost effectiveness.

4.1. Problem detection

The number of usability problems detected is one of the main metrics reported throughout usability testing literature. It is relevant for our studies since this indicates the level of understanding, i.e. if the issues detected by e.g. software development practitioners are usability problems, we argue that understanding is achieved. The level of understanding was examined in Bak et al. (2008) where software managers were asked to describe what, in their view, a usability evaluation is. In that study it was found that SWPs thought of usability evaluation as functionality testing where "The developer tests that the functions work". Respondents believed they were conducting usability evaluations but in reality they were not. In this section we emphasize software development practitioners' (SWP) and users' ability to detect usability problems, which is compared to that of HCI specialists.

The feasibility study of the barefoot approach was conducted in a lab setting and was based on video data analysis. Here we found that the five software development practitioners on average were able to identify just above 48% (# = 24.2, SD = 8.1) of all problems individually and the usability specialist in comparison found 62% (# = 31), see Fig. 4. We also examined the thoroughness of each pair of SWPs and found that they on average are able to identify little more than 71% of all problems (# = 35.7, SD = 5.2).

The follow-up study of the barefoot approach was conducted in an office setting at the case company. Here the three participating SWPs conducted instant data analysis and the three usability specialists conducted video data analysis. Two usability tests (one before and one after revising the system) were conducted in that study and we found similar performances in both tests. The SWPs identified 81% of all problems (# = 33) in the first usability test and 80% (# = 35)

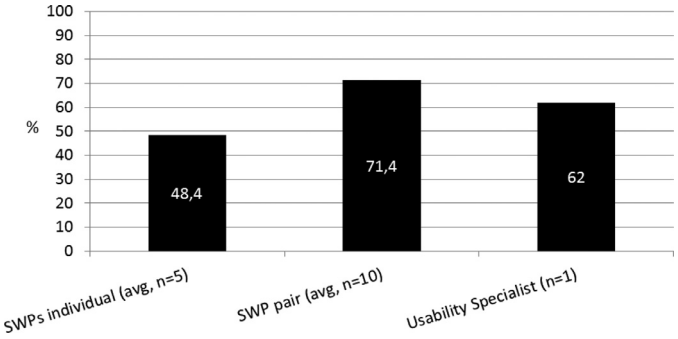

Fig. 4. Identified thoroughness (%) of software development practitioners (SWP) and usability specialist in the first barefoot study conducted in lab settings. SWPs and specialist applied video data analysis (n = number of participants or participant pairs).

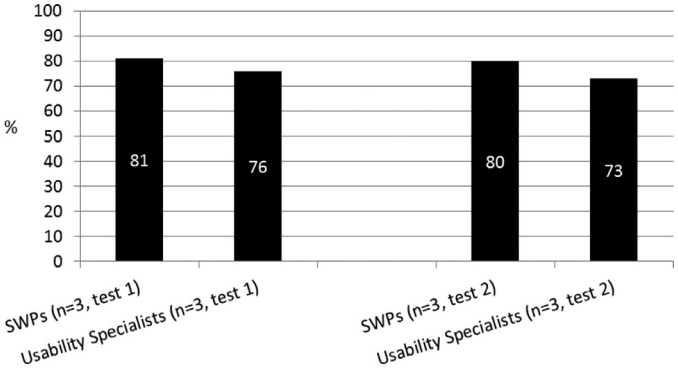

Fig. 5. Identified thoroughness (%) of software development practitioners (SWP) and usability specialists in the follow-up barefoot study conducted in office settings. SWPs applied IDA and specialists applied video data analysis (n = number of participants).

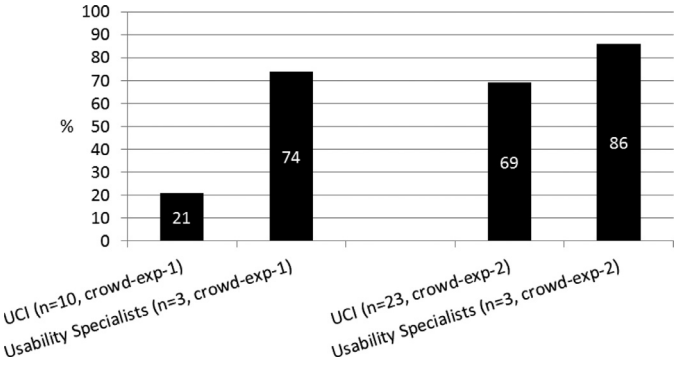

Fig. 6. Identified thoroughness (%) of the UCI crowdsourcing approach and usability specialists conducting video data analysis in the two crowdsourcing experiments (n = number of participants).

in the second, cf. Fig. 5. In comparison the three usability specialists identified 76% of all problems (# = 31) and 73% (# = 32) in test 1 and test 2 respectively. Due to the relatively low number of observations we conducted a Fisher's exact test. This revealed no significant differences between the thoroughness of the SWPs and specialists in the first test (df = 1, $p > 0.7$), which is also the case for the second test (df = 1, $p > 0.6$).

Considering the first crowdsourcing study we found that the 10 student participants in that case found 21% of all problems (# = 13) by applying the UCI method, cf. Fig. 6. In comparison the three usability specialists found 74% (# = 46) through classical video based analysis. Considering the second crowdsourcing study we found that the 23 participants using UCI identified 69% of all problems (# = 29) and that the three specialists doing video based analysis revealed 86 % (# = 36). In the first study a Fishers exact test reveals a significant difference between the crowdsourcing approach of UCI and the conventional laboratory evaluation conducted by specialists (df = 1, $p < 0.001$). In the second study, we did not find a significant difference between these approaches (df = 1, $p > 0.1$).

4.2. Downstream utility

Downstream utility is a measure of the impact that results of usability tests have on an evaluated system (Law, 2006; Sawyer et al., 1996). This is calculated as the committed impact ratio and completed-to-date impact ratio. These ratios are relevant indicators of the level of resistance towards usability testing results. Bak and colleagues found that usability problems are not always accepted by members in the development team, which in turn leads to low priority of fixing identified usability problems (Bak et al., 2008). In the second barefoot study we asked SWPs to do two evaluations in order to measure the downstream utility of this approach.

4.2.1. Committed impact ratio

The committed impact ratio (CIR) is a measure of the problems that software development practitioners commit to fixing before starting to implement a new version of a system. After the SWPs had conducted the first test, they committed to fix 20 problems of the 33 problems they identified. This gives a committed impact ratio of 61%:

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

SWPs in our case study mainly committed to fixing problems based on the factors of resource requirements (problems that were cheap to fix) and coherence to other systems while it did not matter whether a problem was experienced by a single or multiple test users. Severity ratings and length of problem descriptions were less influential. In Bruun and Stage (2012b) further details on this are provided.

4.2.2. Completed-to-date impact ratio

The completed-to-date impact ratio (CDIR) is an expression of the amount of identified usability problems, which have actually been fixed in a revised system version. In the second barefoot study we found that 12 out of the total 33 problems identified during the first test recurred in the revised system. Thus, 21 problems were fixed, which gives a CDIR of 64%:

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

The main reason of why some of the problems recurred in the second version of the system were that the SWPs actually tried to fix most of these, but that these fixes did not work as intended. Additionally, one of the problems was not accepted after occurring in the first test, but was then prioritized after its presence in the second, see Bruun and Stage (2012b) for further details.

4.3. Cost effectiveness

Resource constraint is considered a central obstacle in various studies (Bak et al., 2008; Gulliksen et al., 2004; Gunther et al., 2001; Rosenbaum et al., 2000). This is especially critical in small software development companies as these have limited funding. Cost effectiveness is defined in Hartson et al. (2001) as the combination of a methods ability to detect usability problems and the cost required to do so. Below we measure the average time spent on identifying each usability problem.

In the first crowdsourcing study we studied the cost effectiveness in terms of the average amount of time spent identifying each problem. Looking at the total time spent on preparation and analysis it was found that the laboratory evaluation conducted by specialists required 61 person hours. The UCI method applied by the 10 student

participants required little more than 8 person hours in total. This gave the following cost effectiveness of the methods:

$$\frac{\text{Total time spent}_{UCI}}{\text{No of problems identified}_{UCI}} = \frac{493 \text{ min}}{13 \text{ problems}} = 38 \text{ min.per problem}$$

$$\frac{\text{Total time spent}_{LAB}}{\text{No of problems identified}_{LAB}} = \frac{3663 \text{ min}}{46 \text{ problems}} = 80 \text{ min.per problem}$$

Thus, it required considerably fewer minutes to reveal each problem through UCI compared to the laboratory method applied by specialists.

5. Discussion

The aim of the four case studies was to examine the extent to which barefoot and crowdsourcing evaluations can reduce three main obstacles in adopting usability evaluation methods in practice. These obstacles relate to *limited understanding* of the usability concept, *resistance* towards adopting usability evaluation practices and *resource constraints*. These particular obstacles have been reported since the early 2000s but are still present in recent studies. Overall we found that barefoot evaluations reduced obstacles related to limited understanding and resistance and the crowdsourcing approach is highly effective in reducing the obstacles related to resource constraints. In the following we provide a systematic discussion of the relative strengths and limitations of the barefoot and crowdsourcing approaches in reducing these obstacles.

5.1. Increasing understanding

The level of understanding was examined in Bak et al. (2008) where software managers were asked to describe what a usability evaluation is to them. In that study it was found that 30% emphasized that usability evaluation was about functionality testing, e.g. "The developer tests that the functions work". Thus, these respondents believed they were conducting usability evaluations but in reality they were not. A more detailed approach to determine the level of understanding is to measure the ability to plan and conduct usability evaluations as well as analyzing the data. In this paper we emphasize software development practitioners' (SWP) and users' ability to detect usability problems, which is compared to that of HCI specialists. If the issues detected by software development practitioners are usability problems rather than functionality bugs, we argue that a high level of understanding is achieved.

Findings from our feasibility study on the barefoot approach showed that individual SWPs identified a mean of 48% of all known usability problems in evaluated systems. Additionally, a pair of SWPs revealed 71% on average, i.e. a pair of SWPs outperformed one usability specialist when conducting traditional video based analysis. The follow-up barefoot study showed a similar tendency where three SWPs conducting Instant Data Analysis (IDA) detected 80% of all known problems while three usability specialists identified 74%.

Related work report generally report a lower detection rate than what we found in our barefoot studies. Wright and Monk (1991) for instance found that all student teams in their study on average revealed 33% of all problems. Koutsabasis et al. (2007) found that students identified a mean of 24% of all known problems. The study presented in Frøkjær and Lárusdòttir (1999) shows that students were able to identify 18% of all known problems while detection rate of students in Ardito et al. (2006) is even lower (11%). The study presented in Skov and Stage (2009) compares student performance to that of specialists and show that students identified a mean of 37% of the problems identified by specialists.

It seems surprising that SWPs in our studies were able to outperform usability specialists, especially since they had little or no previous usability testing experience and since they only received 30 hours training. This, however, can be explained through their level of domain knowledge which was higher than that of the specialists who in this case fit the notion of being external consultants. According to Bruce and Morris an inherent problem in applying external consultants is, that they lack domain knowledge (Bruce and Morris, 1994). The importance of domain knowledge is also supported in other studies, e.g. Nielsen's study of usability specialists, non-specialists and double experts (Nielsen, 1992). 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 cooperative usability evaluations (Følstad and Hornbæk, 2010). This 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 indicate that domain knowledge plays a key role in the identification of usability problems. This indicates an advantage of the barefoot evaluation approach over separate unit and outsourcing approaches where usability specialists are distant from the team that develops the software.

In comparison we found that the crowdsourcing approach reveal a lower detection rate. Findings from our first crowdsourcing study show that a laboratory evaluation conducted by specialists reveals significantly more problems than the User reported Critical Incident (UCI) method. In that study UCI uncovered 21% of all known problems. However, participants in the second crowdsourcing study identified 69% of all known problems, which was not significantly different from the laboratory test. Yet, these detection rates are still lower than those found during our studies of the barefoot approach. The observed differences in detection rates between the barefoot and crowdsourcing approaches could be attributed to the amount and type of training. The SWPs in the barefoot studies received 30 hours of in-person training compared to the written instructions submitted remotely to participants in the crowdsourcing studies. There are relatively few research papers on UCI. Our findings from the first crowdsourcing study are comparable to those found in Andreasen et al. (2007). In our study we found a thoroughness of 28% of the problems found in the conventional laboratory condition while Andreasen et al. (2007) found 37%. Other earlier studies revealed remarkably different results where the UCI method in e.g. Castillo et al. (1998) revealed a thoroughness of 76%. These earlier findings are similar to the thoroughness of 69% found in our second crowdsourcing experiment.

The study presented in Høegh et al. (2006) examined how to increase practitioners' awareness of usability problems (Høegh et al., 2006). This was done by introducing different formats for providing feedback from usability tests. One of the feedback formats was to let the practitioners gain first hand observations from the test sessions, i.e. to further involve them in the process. This was superior to just providing a written report. The barefoot approach takes this a step further as the SWPs plan and conduct evaluations as well as analyzing data. As shown in the above, this can lead to a high level of understanding. This, however, also leads to an apparent downside of crowdsourcing evaluations as these provide limited first hand insights of the users during interaction. Although the second crowdsourcing study led to a relatively high detection rate of usability problems, we still argue that the level of understanding will not be increased to the same extent as when applying barefoot evaluations. In sum, we argue

that the barefoot approach reduces the obstacles related to limited understanding due to the first hand insights gained through training of SWPs and the following conduction of usability tests. On its own, however, the crowdsourcing approach will not alleviate this obstacle as it is only the users who receive training and not those responsible for improving the systems.

5.2. Reducing resistance

The resistance obstacle concerns developers' the acceptance of usability testing results. In Bak et al. (2008) it is mentioned that usability problems are not always accepted by members in the development team (Bak et al., 2008), which in turn leads to low priority of fixing identified usability problems.

Downstream utility denotes the extent to which results from usability evaluations impact the usability of a system (Law, 2006). Downstream utility is thus an indicator of resistance towards results obtained through usability testing where, e.g. a low level of downstream utility can be caused by a high level of resistance.

In the follow-up study of the barefoot evaluation approach, we found that the SWPs committed to fixing most of the identified problems and that they prioritized these based on the factors of resource requirements and coherence to other systems. Additionally, the practitioners managed to eliminate most of the problems, which resembles the downstream utility found in another organizational setting where usability practices had already been established, cf. Hertzum (2006). These findings, combined with the fact that the practitioners identified a considerable amount of problems, indicate that the barefoot evaluations caused practitioners to accept results from usability evaluations as well as prioritize fixing problems. This deviates from the typical developer mindset described in the literature, cf. Ardito et al. (2011), Bak et al. (2008). This finding may be explained by the understanding that follows from the direct observation of users as this provides first hand insights into the usability problems experienced, an effect which is also noted in Høegh et al. (2006).

The two studies concerning crowdsourcing evaluations indicate that UCI enables users to identify usability problems. However, the outcome is still only a list of problems on which the software development practitioners need to base their improvements. In this case the software development practitioners do not observe the evaluations. This corresponds to the approach where software development practitioners receive a written list of usability problems, which is the most widely used feedback format (Høegh et al., 2006). Due to the fact that software development practitioners do not observe the users during interaction, this understanding would arguably be compromised when applying crowdsourcing evaluations and, in turn, lead to a lower level of downstream utility. However, further studies are needed to validate this claim.

Finally, the practitioners in the follow-up barefoot study managed to eliminate most of the problems found in the initial version of the system. However, it was also found that the second version introduced a considerable amount of new problems. This behavior is recognized 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 points out, then it is equally important to tell the practitioners what to do and not just what is wrong within an interface (Wixon, 2003). Thus, in the future it would be crucial to provide such practitioners with training in interaction design to further increase the impact of usability evaluations.

5.3. Reducing costs

Resource constraint is reported in several studies to be a central obstacle against the adoption of usability testing practices (Bak et al., 2004; Gulliksen et al., 2004; Gunther et al., 2001; Rosenbaum et al., 2000). This is especially critical in small software development companies as these have limited funding. In Hartson et al. (2001) cost effectiveness is defined as the combination of an evaluation method's ability to detect usability problems and the cost required to do so. In our studies we derived cost effectiveness as the average time spent on identifying each usability problem.

The first crowdsourcing study shows that the UCI method requires considerably fewer resources compared to traditional laboratory testing based on video analysis. Findings show that the total time required to prepare evaluations and analyze results required 61 person hours in case of the traditional laboratory test while the UCI method required little more than 8 person hours. In relation to crowdsourcing, Doan and colleagues mention that the challenge of combining user input is a relatively simple task when users provide quantitative data such as numeric ratings, as this can be done automatically (Doan et al., 2011). Qualitative input such as free form text requires a higher degree of manual labor (Doan et al., 2011) which could compromise the aim of lowering the amount of required resources through the user driven approach. The usability problems reported by the users are qualitative in nature. However, when considering the cost effectiveness it is shown that the UCI required less than 50% of the time to uncover each problem compared to the laboratory method while also providing around 50% of the critical problems. This also included time spent on matching and filtering valid and invalid problem descriptions. In relation to resource demands, all instructions in our crowdsourcing experiments were conveyed online in written form, which shows that larger groups of users can indeed receive minimalist training and successfully identify usability problems using few resources.

Considering barefoot evaluations it was found that the practitioners were able to identify a large amount of usability problems after receiving 30 hours of training. This shows that such practitioners may obtain considerable competences in what may seem to be a short time frame. On the other hand it can be difficult to overcome the obstacle of high resource constraints 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 testing. In the long run, however, it can be argued that barefoot evaluations would require less resources as the hourly rates of external consultants usually are higher than that of internal employees. A study by Bruce and Morris (1994) supports this by mentioning that in-house designers are less expensive to use compared to out-house designers. An additional consideration is that the practitioners in the barefoot experiments had various job responsibilities as e.g. system developers, test managers and project managers. This means that they have other tasks than just conducting usability tests, and that when they spend time on usability testing they cannot fulfill other responsibilities such as implementation and planning activities. These other tasks must then be completed at a different point in time. This resembles the critique raised against Deng Xiaopings suggestion of letting China's barefoot doctors gradually "put on shoes" by improving their medical skills, as this moved their responsibilities further away from that of agricultural production (Daqing and Unschuld, 2008).

5.4. Complementarity of methods

Based on the above discussion it seems that barefoot evaluations are superior to the crowdsourcing evaluations in terms of reducing obstacles of limited understanding and resistance while crowdsourcing is beneficial in overcoming the obstacle of resource constraints. In the context of small software development companies, the barefoot approach is highly beneficial as these companies often have no available HCI competences. Since crowdsourcing evaluations provide training for the users only, it is not suitable to apply that approach in such a context as it requires development team members that are able

to provide user training and are able to analyze user reports. Nevertheless, the cost effectiveness of that approach is highly relevant for small companies. As mentioned in the introduction of this paper, then there are several causes for the limited adoption of usability practices. Although the obstacles of *resource constraints, limited understanding* and *resistance* have been known for over a decade, these obstacles are still present. We believe that the barefoot and crowdsourcing approaches are highly complementary and that they in concert are able to reduce these obstacles. As an example, a small software development company could begin adopting usability testing practices by following the barefoot approach. This will increase competences of development team members, hereby increasing awareness and reducing resistance towards usability testing results. Having established basic competences, the actual conduction of tests could then be based on crowdsourcing, which is a highly cost effective alternative to classical usability testing methods.

5.5. Limitations

There is a range of limitations that needs to be addressed in future work. First of all, we emphasized the three main obstacles reported in HCI literature. However, there exists a broad range of other obstacles in relation to adopting User Centered Design (UCD) practices in real-world ICT development. We elaborate on these in the next subsection. This is followed by a discussion of limitations regarding the empirical method applied in the four case studies.

5.5.1. Context of design

Svanæs and Gulliksen made a distinction between projects related obstacles and obstacles originating from the context of design, i.e. in a scope outside project boundaries (Svanæs and Gulliksen, 2008). They identify several of such project boundary aspects, which could pose obstacles to successful adoption of UCD methods. Four examples of these obstacles relate to: *internal factors in the developer organization*, *internal factors in the client organization*, *customer–developer legal relationships* and *organizational stability*. While we emphasized project related obstacles, we in the following point to obstacles from the context of design, which cannot be handled through the barefoot and crowdsourcing evaluations.

Obstacles related to *internal factors in the developer organization* could for instance be strategic decisions made by management outside the development team, which conflicts with UCD work. Svanæs and Gulliksen mention lack of management support to be one of the main constraints to UCD, which is also supported in Gulliksen et al. (2004), Schaffer (2004), Venturi et al. (2006). This can lead to a low prioritization of usability matters (Gulliksen et al., 2004). Gulliksen et al. (2004) note that emphasizing the importance of UCD in the education of software developers and other stakeholders, such as management, could be a means to decrease this obstacle. In the two barefoot studies we had initial support from the upper management to go ahead with the training activities. This provided a clear advantage in our case as an obstacle of low management support was cleared out.

Internal factors in the customer organizations can also pose obstacles of external origin. In relation to this Svanæs and Gulliksen (2008) mention limited or no access to users, which is also supported in Ardito et al. (2011), Bak et al. (2008), Poltrock and Grudin (1994), Wilson et al. (1997). In some situations customer organizations prohibit access to users (Svanæs and Gulliksen, 2008). Poltrock and Grudin mention that overcoming this obstacle requires stakeholder determination to prioritize a higher level of usability to an extent that imposes change (Poltrock and Grudin, 1994), and, affecting this determination necessitates knowledge of what is required to achieve usability (Poltrock and Grudin, 1994). However, there can still be obstacles related to user access, even in situations where the customer organization has granted access to these. Wilson et al. (1997) mention a case where users were too busy to participate, which resulted in users not showing up for planned activities. In this case access to users becomes a question of time and logistic practicality. The crowdsourcing approach could reduce this obstacle as users do not need to travel long distances to participate in an evaluation. However, if access to users is prohibited by company policies, such as the example provided in Svanæs and Gulliksen (2008), it is questionable that the barefoot and crowdsourcing approaches will be a solution.

Organizational stability is also an aspect that potentially could influence UCD, e.g. in a situation where the company is bought by a competitor and UCD efforts and documentation in current projects ends up "in a drawer" due to change in management (Svanæs and Gulliksen, 2008). This aspect overlaps that of management support. Svanæs and Gulliksen (2008) mention that in most cases it is impossible to change aspects such as those mentioned above, but the risk in some cases can be reduced, e.g. by training developers to apply UCD methods. The obstacle of organizational instability where UCD documentation ends up in a drawer could be reduced by focusing less on usability artifacts and more on learning and knowledge transfer (Sanæs and Gulliksen, 2008), which is in line with the barefoot approach. However, research on how it influences this particular obstacle remains to be studied.

Additionally, Svanæs and Gulliksen mention obstacles related to *contractual and tender issues*. They give an example from a case study in which the tender process required all system requirements to be specified before the call for tender. This in turn left no room for iterating on requirements once the contract was signed and the users were not involved until completion of the system a couple of years later. Overcoming this obstacle requires determination from the stakeholders in the customer organization to prioritize UCD activities as mentioned by Poltrock and Grudin (1994).

In general, the above types of obstacles originate from outside the projects and it can, thus, be close to impossible to overcome these from within the frame of a given project (Svanæs and Gulliksen, 2008), even if the project utilizes barefoot and crowdsourcing evaluations. Schaffer provides valid pointers on how to overcome obstacles from the context of design, cf. Schaffer (2004).

5.5.2. Empirical method

Our barefoot studies showed that the software development practitioners were able to remove most usability problems, which led to a high level of downstream utility. However, a considerable amount of new problems were identified in the updated version of the system. As argued by Nielsen, it is typical that new problems occur in a revised interface design (Nielsen, 1993), but this could be reduced if the practitioners had received training in interaction design. As Wixon points out, then it is equally important to tell the practitioners what to do and not just what is wrong in a user interface (Wixon, 2003). Therefore, we believe a natural continuation of our work would be to train software development practitioners in interaction design. It would be interesting to study how this affects the extent of new usability problems being introduced in revised interface designs.

Furthermore, Doan and colleagues emphasize the challenge of recruiting and retaining users in relation to crowdsourcing (Doan et al., 2011). In our crowdsourcing studies we recruited university students as participants. As employees at the same university we had knowledge of the best communication channels to use to get in contact with students. In our cases this was through the respective semester secretaries. Such knowledge, however, may not always be available. Additionally, as the students came from the same university, this could have had an effect on retaining the users, e.g. they were motivated to participate as we were part of the same organization. This is not always the case in practice, which is why it could be interesting to conduct further studies of how knowledge of information channels and closeness of relationships affect recruiting and retaining users in the crowdsourcing approach. That said, there are several other

alternatives for conducting evaluations based on crowdsourcing. Services such as usertesting.com, trymyui.com and usabilla.com deliver user feedback through commented videos. More recently, Heintz et al. (2014) developed the Pdot tool. Pdot enables digital online participatory design where development teams get graphical and annotated feedback on designs.

Finally, the four studies involve matching of usability problems. Hornbæk and Frøkjær (2008) argue that the evaluator effect is influenced by similarities in the criteria used for matching problems. If there are no clear criteria on how to compare problems during the matching process, this could lead to a lower agreement between evaluators. In Hornbæk and Frøkjær (2008) it is also found that matching problems in teams provides a higher agreement compared to individual matching as well as a greater evaluator satisfaction on the matches made. During our studies we applied the same structured formats for describing usability problems to ensure these were described at similar levels of granularity. Additionally, we did conduct problem matching in teams to counteract evaluator effect inflation. Although findings indicate that these measures counteract the evaluator effect (Hornbæk and Frøkjær, 2008), this will still be substantial in these types of studies.

6. Conclusions

We have reported and discussed findings from four case studies exploring the feasibility of barefoot and crowdsourcing usability evaluations. Our studies showed that these approaches can reduce the three critical obstacles related to *resource constraints, limited understanding* of the usability concept and methods as well as *resistance* towards adopting usability practices. These obstacles have been known for more than a decade, but are still present, especially within small software development companies. The barefoot and crowdsourcing approaches each have relative strengths and limitations in reducing these obstacles.

We found that barefoot evaluations are well suited to overcome obstacles related to limited understanding and resistance while crowdsourcing is highly effective in overcoming the obstacle of resource constraints. Software development practitioners received 30 hours of training in usability testing and were able to detect and fix a considerable number of usability problems. Crowdsourcing evaluations are based on minimalist training of end users to conduct evaluations and report problems remotely. Given that crowdsourcing evaluations provide training for the users only, it is not suitable to apply this approach in contexts with no HCI competences. It requires development team members that can train end users and analyze user reports. Nevertheless, the cost effectiveness of crowdsourcing evaluations is highly relevant for small companies. We believe that the barefoot and crowdsourcing approaches are highly complementary and that they in concert can be effective in reducing aforementioned obstacles.

As a continuation of the work presented in this paper there are four areas, where it would be particularly relevant to conduct further research. Firstly, it would be relevant to conduct empirically based studies of downstream utility in relation to the crowdsourcing approach, but also systematic studies of the cost effectiveness of the barefoot approach. Secondly, it would be relevant to further support practitioners through training in interaction design and then study how this affects downstream utility. Thirdly, it would also be relevant to conduct field experiments with more practitioners and companies participating to increase generalizability of findings in relation to the barefoot approach. Fourthly, we find a need for studying sustainability of the barefoot approach imposed in the case company, which could be accomplished through further longitudinal studies. This also applies in case of the crowdsourcing approach. At the time of writing, a 20 month period has passed without any research activities in the case company applying the barefoot approach. In that period practitioners have initiated three usability tests on their own, this indicates sustainability of that approach.

Acknowledgments

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

References

Andreasen, M.S., Nielsen, H.V., Schrøder, S.O., Stage, J., 2007. What happened to remote usability testing? An empirical study of three methods. In: Proceedings of the CHI. ACM Press.
Ardito, C., Costabile, M.F., De Angeli, A., Lanzilotti, R., 2006. Systematic evaluation of e-learning systems: an experimental validation. In: Proceedings of the NordiCHI. ACM Press.
Ardito, C., Buono, P., Caivano, D., Costabile, M.F., Lanzilotti, R., Bruun, A., Stage, J., 2011. Usability evaluation: a survey of software development organizations. In: Proceedings of the of SEKE. Knowledge Systems Institute Graduate School.
Bak, J.O., Nguyen, K., Risgaard, P., Stage, J., 2008. Obstacles to usability evaluation in practice: a survey of software development organizations. In: Proceedings of the NordiCHI. ACM Press.
Bosenick, T., Kehr, S., Kühn, M., Nufer, S., 2007. Remote usability tests: an extension of the usability toolbox for online-shops. In: Proceedings of the UAHCI. Springer.
Bruce, M., Morris, B., 1994. Managing external design professionals in the product development process. Technovation 14 (9), 585–599 (Elsevier).
Bruun, A., 2010. Training software development practitioners in usability testing: an assessment acceptance and prioritization. In: Proceedings of the NordiCHI. ACM Press.
Bruun, A., Gull, P., Hofmeister, L., Stage, J., 2009. Let your users do the testing: a comparison of three remote asynchronous usability testing methods. In: Proceedings of the CHI. ACM Press.
Bruun, A., Stage, J., 2011. Training software development practitioners in usability evaluations: an exploratory study of cross pollination. In: Proceedings of the PUX.
Bruun, A., Stage, J., 2012a. The effect of task assignments and instruction types on remote asynchronous usability testing. In: Proceedings of the CHI. ACM Press.
Bruun, A., Stage, J., 2012b. Training software development practitioners in usability testing: an assessment acceptance and prioritization. In: Proceedings of the OzCHI. ACM Press.
Castillo, J.C., 1997. The User-Reported Critical Incident Method for Remote Usability Evaluation (Master thesis). Virginia Polytechnic Institute and State University.
Castillo, J.C., Hartson, H.R., Hix, D., 1998. Remote usability evaluation: can users report their own critical incidents? In: Proceedings of the CHI. ACM Press.
Daqing, Z., Unschuld, P.U., 2008. China's barefoot doctor: past, present, and future. Lancet 372 (9653), 1865–1867 (Elsevier).
Doan, A., Ramakrishnan, R., Halevy, A.Y., 2011. Crowdsourcing systems on the World-Wide Web. Commun. ACM 54 (4), 86–96 (ACM).
Felder, R.M., Silverman, L.K., 1988. Learning and teaching styles in engineering education. Eng. Educ. 78 (7), 674–681.
Frøkjær, E., Lárusdòttir, M.K., 1999. Prediction of usability: comparing method combinations. In: Proceedings of the IRMA. Idea Group Publishing.
Følstad, A., Hornbæk, K., 2010. Work-domain knowledge in usability evaluation: experiences with cooperative usability testing. J. Syst. Softw. 83 (11), 2019–2030 (Elsevier).
Gulliksen, J., Boivie, I., Persson, J., Hektor, A., Herulf, L., 2004. Making a difference: a survey of the usability profession in Sweden. In: Proceedings of the NordiCHI. ACM Press.
Gunther, R., Janis, J., Butler, S., 2001. The UCD Decision Matrix: How, When, and Where to Sell User-Centered Design into the Development Cycle. http://www.ovostudios.com/upa2001/ (accessed 21.11.14).
Hartson, H.R., Andre, T.S., Williges, R.C., 2001. Criteria for evaluating usability evaluation methods. Int. J. Hum.–Comput. Interact. 13 (4), 373–410 (Taylor & Francis).
Häkli, A., 2005. Introducing User-Centered Design in a Small-Size Software Development Organization. Helsinki University of Technology.
Heintz, M., Law, E.L.C., Govaerts, S., Holzer, A., Gillet, D., 2014. Pdot: participatory design online tool. In: Proceedings of the CHI. ACM Press.
Hertzum, M., 2006. Problem prioritization in usability evaluation: from severity assessments toward impact on design. Int. J. Hum.–Comput. Interact. 21 (2), 125–146 (Taylor & Francis).
Hornbæk, K., 2010. Dogmas in the assessment of usability evaluation methods. Behav. Inform. Technol. 29 (1), 97–111 (Taylor & Francis).
Hornbæk, K, Frøkjær, E, 2008. A study of the evaluator effect in usability testing. Hum.–Comput. Interact. 23 (3), 251–277 (Taylor & Francis).
Howarth, J., 2007. Supporting Novice Usability Practitioners with Usability Engineering Tools. Virginia Polytechnic Institute & State University Blacksburg.
Howarth, J., Andre, T.S., Hartson, R., 2007. A structured process for transforming usability data into usability information. J. Usabil. Stud. 3 (1), 7–23.
Høegh, R.T., Nielsen, C.M., Overgaard, M., Pedersen, M.B., Stage, J., 2006. A qualitative study of feedback from usability evaluation to interaction design: are usability reports any good? Int. J. Hum.–Comput. Interact. 21 (2), 173–196 (Lawrence Erlbaum Associates).
ISO 9241-11, 1998. Ergonomic Requirements for Office Work with Visual Display Terminals (CDTs). Part 11. Guidance on Usability. International Organization for Stadardization.

Juristo, N., Moreno, A.M., Sanchez-Segura, M.I., 2007. Guidelines for eliciting usability functionalities. IEEE Trans. Softw. Eng. 33 (11), 744–758 (IEEE Computer Society Press).
Kjeldskov, J., Skov, M.B., Stage, J., 2004. Instant data analysis: conducting usability evaluations in a day. In: Proceedings of the NordiCHI. ACM Press.
Koutsabasis, P., Spyrou, T., Darzentas, J.S., Darzentas, J., 2007. On the performance of novice evaluators in usability evaluations. In: Proceedings of the PCI.
Law, E., 2006. Evaluating the downstream utility of user tests and examining the developer effect: a case study. Int. J. Hum.–Comput. Interact. 21 (2), 147–172 (Taylor & Francis).
Millen, D.R., 1999. Remote usability evaluation: user participation in the design of a web-based email service. ACM SIGGROUP Bull. 20 (1), 40–45 (ACM Press).
Molich, R., 2000. User-Friendly Web Design. Ingeniøren Books.
Molich, R, Dumas, J, 2008. Comparative usability evaluation (CUE-4). Behav. Inform. Technol. 27 (3), 263–281 (Taylor & Francis).
Nielsen, J., 1992. Finding usability problems through heuristic evaluation. In: Proceedings of the CHI. ACM Press.
Nielsen, J., 1993. Iterative user-interface design. Computer 26 (11), 32–41 (IEEE).
Nielsen, J., 1994. Usability inspection methods. In: Proceedings of the CHI. ACM Press.
Poltrock, S.E., Grudin, J., 1994. Organizational obstacles to interface and development: two participant – observer studies. ToCHI 1 (1), 52–80 (ACM Press).
Prince, M.J., Felder, R.M., 2006. Inductive teaching and learning methods: definitions, comparisons, and research bases. Eng. Educ. 95 (2), 123–138.
Rosenbaum, S., Rohn, J.A., Humburg, J., 2000. A toolkit for strategic usability: results from workshops, panels, and surveys. In: Proceedings of the CHI. ACM Press.
Rubin, J., Chisnell, D., 2008. Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests, second ed. John Wiley & Sons.
Sawyer, P., Flanders, A., Wixon, D., 1996. Making a difference – the impact of inspections. In: Proceedings of the CHI. ACM Press.
Schaffer, E., 2004. Institutionalization of Usability: A Step-by-Step Guide. Pearson.
Scholtz, J., 1999. A case study: developing a remote, rapid and automated usability testing methodology for on-line books. In: Proceedings of the HICSS. IEEE.
Scholtz, J., Downey, L., 1998. Methods for identifying usability problems with web sites. In: Proceedings of the Engineering for Human–Computer Interaction. Springer.
Skov, M.B., Stage, J., 2005. Supporting problem identification in usability evaluations. In: Proceedings of the OZCHI. ACM Press.
Skov, M.B., Stage, J., 2009. Training software developers and designers to conduct usability evaluations. Behav. Inform. Technol. 31 (4), 425–435 (Taylor & Francis).
Steves, M.P., Morse, E., Gutwin, C., Greenberg, S., 2001. A comparison of usage evaluation and inspection methods for assessing groupware usability. In: Proceedings of the CSCW. ACM Press.
Svanæs, D., Gulliksen, J., 2008. Understanding the context of design: towards tactical user centered design. In: Proceedings of the NordiCHI. ACM Press.
Thompson, J.A., 1999. Investigating the Effectiveness of Applying the Critical Incident Technique to Remote Usability Evaluation (Master thesis). Virginia Polytechnic Institute and State University Blacksburg.
Venturi, G., Troost, J., Jokela, T., 2006. People, organizations, and processes: an inquiry into the adoption of user-centered design in industry. Int. J. Hum.–Comput. Interact. 21, 219–238 (Taylor & Francis).
Waterson, S., Landay, J.A., Matthews, T., 2002. In the lab and out in the wild: remote web usability testing for mobile devices. In: Proceedings of the CHI. ACM Press.
West, R., Lehman, K.R., 2006. Automated summative usability studies: an empirical evaluation. In: Proceedings of the CHI. ACM Press.
Wilson, A., Bekker, M., Johnson, P., Johnson, H., 1997. Helping and hindering user involvement – a tale of everyday design. In: Proceedings of the CHI. ACM Press.
Winckler, M.A.A., Freitas, C.M.D.S., de Lima, J.V., 1999. Remote usability testing: a case study. In: Proceedings of the OzCHI. ACM Press.
Winckler, M.A.A., Freitas, C.M.D.S., de Lima, J.V., 2000. Usability remote evaluation for www. In: Proceedings of the CHI. ACM Press.
Wixon, D., 2003. Evaluating usability methods: why the current literature fails the practitioner. Interactions 10 (4), 28–34 (ACM Press).
Wright, P.C., Monk, A.F., 1991. The use of think-aloud evaluation methods in design. ACM SIGCHI Bull. Arch. 23 (1), 55–57 (ACM Press).

Anders Bruun is assistant professor at the Department of Computer Science, Aalborg University. His research interests include methods for usability evaluation, interaction design and HCI in relation to health informatics. He achieved his Ph.D. in January 2013. He formerly worked as a User Experience consultant in a Danish software company where he was responsible for designing and evaluating user interfaces in small and large scale web projects ranging from 500 to 15,000 person hours. He is member of the editorial board of the American Journal of Heath Informatics.

Jan Stage is full professor in information systems and human-computer interaction at the Department of Computer Science, Aalborg University (Denmark). He obtained a PhD in Computer Science from University of Oslo in 1989. His research interests include methods for usability evaluation, interaction design and usability evaluation of mobile systems, object-oriented analysis and design, and prototyping. He is leading a national research project on usability engineering in software development and involved in European research on usability. He is an associate editor of Behaviour and Information Technology, member of the editorial boards of Journal of Database Management and International Journal of Mobile Human Computer Interaction, guest editor for International Journal of Human-Computer Interaction in 2006 and Journal of Systems and Software in 2010, member of the program committees of the Interact 2009 and NordiCHI 2010 conferences, and reviewer for several journals and conferences in the HCI area.