Distributed usability evaluation: enabling large-scale usability evaluation with user-controlled Instrumentation

Conference Paper · January 2010

DOI: 10.1145/1868914.1868932 · Source: DBLP

Erik Frøkjær
University of Copenhagen
41 PUBLICATIONS **1,396** CITATIONS

Distributed Usability Evaluation: Enabling Large-scale Usability Evaluation with User-controlled Instrumentation

Lars Christensen & Erik Frøkjær
Department of Computing, University of Copenhagen
Njalsgade 128, building 24, 5th floor, DK-2300 Copenhagen S, Denmark
{larsc,erikf}@diku.dk

ABSTRACT
We present DUE (Distributed Usability Evaluation), a technique for collecting and evaluating usability data. The DUE infrastructure involves a client-server network. A client-based tool resides on the workstation of each user, providing a screen video recording, microphone input of voice commentary, and a window for a severity rating. The idea is for the user to work naturalistically, clicking a button when a usability problem or point of uncertainty is encountered, to describe it verbally along with illustrating it on screen, and to rate its severity. These incidents are accumulated on a server, providing access to an evaluator (usability expert) and to product developers or managers who want to review the incidents and analyse them. DUE supports evaluation in the development stages from running prototypes and onwards. A case study of the use of DUE in a corporate environment is presented. The study indicates that the DUE technique is effective in terms of low bias, high efficiency, and clear communication of usability issues among users, evaluators and developers. Further, DUE is supporting long-term evaluations making possible empirical studies of learnability.

Author Keywords
Usability evaluation, instrumentation, automation, remote, distributed, screen video, voice commentary, case study, software industry, evaluator effect, think-aloud, beta test, learnability.

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation (e.g., HCI): User Interfaces–Evaluation/Methodology; D.2.2 Software Engineering: Design Tools and Techniques–User Interfaces.

INTRODUCTION
Usability evaluations in industry are limited by a number of challenges, for instance: evaluations are very labor demanding, it can be difficult to establish clear goals for evaluations, and the communication and prioritization necessary for transforming redesign proposals into reality are complicated [20,23].

We have tried to ease some of these difficulties by designing a technique called Distributed Usability Evaluation (DUE) with the following objectives:

- Support the detection of usability problems covering aspects of effectiveness, efficiency and satisfaction.
- Minimize the evaluation workload for users, evaluators and developers.
- Enable evaluation of short- and long-term usage.
- Strengthen the quality and validity of identified usability problems, for example by enabling realistic user-context and large-scale evaluation with many users.
- Strengthen the communication of usability problems between users, evaluators and developers.

To realize the objectives it was clear that an instrumentation tool was needed. We wanted automated screen video capture of user interaction and recording of voice commentary, while the user was using the evaluated application. Like in beta testing, this enabled us to have the user in charge of problem detection, without quite losing the information-rich context known from think-aloud evaluations.

To investigate the DUE technique we carried out an extensive case study. Here DUE was used to evaluate the usability of production software under active development in a business environment with 16 users during one month.

The next section summarizes the key challenges that served as background for our design. Then we present DUE with its Instrumentation Framework and how it can be used by users, evaluators and developers. Next follows a section describing the case study and the strategy for data analysis. Finally, we summarize and discuss the main results.

BACKGROUND
This section will summarize and briefly discuss some of the key challenges in usability evaluation, which we aim to address by DUE.

Measuring Usability

A definition of usability, as the one given by ISO [15], gives us the initial step towards analyzing usability in applications. The next step, identifying and implementing reliable measures of usability, remains a challenge.

Hornbæk [10] points out that usability cannot be measured directly and as a result of this, we find aspects of usability to measure and draw conclusions upon. Frøkjær et al. [4] make it clear that measures used to identify usability problems often do not take into account all three aspects of usability, i.e. effectiveness, efficiency and satisfaction; and a strong correlation between these aspects are often wrongly assumed. Identifying reliable measures remain an open-ended and commonly discussed challenge. Really useful measures cannot be identified at a general level, but have to be based on an understanding of the use situations within the particular application domain [4]. Users often have this understanding.

Instrumentation and Automation

Instrumentation and automation within software evaluation have received some attention as a means to collect measures and supplement software evaluation, but have according to Ivory & Hearst [16] been underexplored.

Early work by Hartson et al. [2,8,9] indicated that many challenges in usability evaluation could be met by using semi-instrumented remote evaluation and critical incident identification by users. However, Hartson et al. did not demonstrate the effectiveness of their approach in field studies.

Recently instrumentation has received renewed focus as a means of collecting measures. Kim et al. [19] shows us that instrumentation can be a powerful tool to help detection of usability problems, especially when users are more involved than typically seen. User initiated events are seen as useful measures. Unfortunately the type of instrumentation used by Kim et al. requires program intrusion of the evaluated application, which limits practical usage in many situations.

Evaluation Tools and Workload

Bateman et al. [1] recognize that setting up application intrusive instrumentation is a time-consuming task requiring technical insight by the evaluators. They argue that benefits can be achieved by being less intrusive; and propose a more intuitive way of adding instrumentation to an application.

Other tools for usability evaluation, which offer some degree of instrumentation and are fully non-intrusive to the application, such as TechSmith's Morae software, also support the task of usability evaluation. Here the evaluator is still left with the responsibility for problem detection and, as Howarth et al. [14] point out, usability evaluation is a tedious task even with this kind of software.

Emerging services, like for instance www.usertesting.com [26], offer internet-based remote think-aloud testing of websites. These services make it fast and easy to set up think-aloud tests and find participants, but many of the limitations and labour demanding tasks related to think-aloud testing still remains, e.g. to extract, analyse and consolidate information from the videos.

Thus, in spite of the assistance from various tools, usability evaluation in general is still a tedious task for evaluators, and often also for the developers. It involves extensive work to identify measures, set up evaluations with or without instrumentation, collect large amounts of data, analyse data, detect problems among the often vague measures, and communicate the problems [20,23].

The Evaluator Effect and Insufficient Analysis

The evaluator effect denotes that usability evaluators identify substantially different sets of usability problems when applying the same evaluation technique on the same application [17]. Facing large amounts of usability data, some of them vaguely documented, can make evaluator assessments complicated and unclear, thus increasing the evaluator effect.

Poor or insufficient analysis of usability data in practice, as observed by Nørgaard & Hornbæk [22], is partly a result of the complexity of the evaluator's work when handling large amounts of usability data.

These difficulties might be eased by tool support and stronger involvement of the users.

Communication

The large amount of complicated data raises a huge communication challenge between people with quite different backgrounds: users with domain knowledge, developers with their technical insight, and evaluators with usability insight. Actually, this challenge is so large that it is often avoided in practice by more or less leaving out some of the stakeholders. However, studies indicate that involving developers, managers as well as users in the evaluation process can improve results [5,6,11,12].

One possibility of strengthening the communication among stakeholders is to support the identification and recording of critical usability data in order to better understand and establish relationships among usability data [14].

Validity

The large workload on primarily the evaluators is probably the main reason for carrying out usability evaluations with few users over short periods of time. This makes it difficult to ensure external validity. The few users can make it doubtful that results are representative for all of the user population [3,25]. Further, a short evaluation period will only give snapshot insight in usability problems leaving out the users' learning curve [7].

To strengthen the ecological validity, the circumstances of the usability evaluations should be as realistic as possible, while keeping attention to the risk of introducing disturbing experimental factors [24].

DISTRIBUTED USABILITY EVALUATION (DUE)

The Instrumentation Framework

We have built an Instrumentation Framework, consisting of an Instrumentation Client and an Instrumentation Server, to realize the instrumentation and process automation necessary for meeting the DUE design objectives.

When doing a DUE evaluation the Instrumentation Client is installed on the computer of each participating user. The only requirements are that the computer must have a certain cpu power, a microphone, and a reasonable fast Internet connection. The Instrumentation Client manages user evaluation sessions. When an evaluation session ends, the Instrumentation Client transfers all collected data to the Instrumentation Server via the Internet.

All collected data are processed and stored on the Instrumentation Server for future use. The Instrumentation Server offers a web interface for evaluators and developers where they can log in and fulfil their role in the DUE workflow. See Figure 1 for the workflow of the four roles involved in DUE. These roles will be described in the following sections.

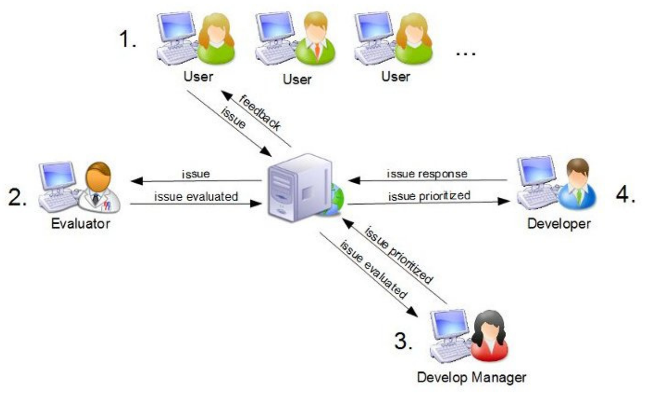


Figure 1. Workflow in DUE.
(1) The user starts with reporting issues, (2) the evaluator assesses these, (3) the development manager prioritizes, and (4) the developers respond. Finally the user receives feedback (1).

The User Role

The instrumentation in DUE is centered on video recorded from the user's screen and voice recorded from a microphone placed at the screen, while the user is using the application. The recording is continuously running while the Instrumentation Client is active, making it unnecessary to reproduce issues after discovery.

Instead of focusing on pre-selected measures and collecting large amounts of instrumented data from the application, which have to be analysed thoroughly to identify usability issues, the focus in DUE is shifted to the user, who is given the responsibility to detect the usability issues. Thereby the primary instrumentation lays in the recorded video of the user's screen, the recording of the user's explanation, sound from surroundings, plus the collection of timestamps, and severity ratings of the user reported issues.

The user is instructed to have focus on elements which can have correlation to effectiveness, efficiency and satisfaction, but most importantly to use common sense and domain knowledge to report when something seems troublesome in actual use.

In practice, when experiencing something as troublesome, the user can press a button to report and assess the severity of the issue (red, yellow or green) in the small Instrumentation Client visible on-screen, Figure 2 and 3.

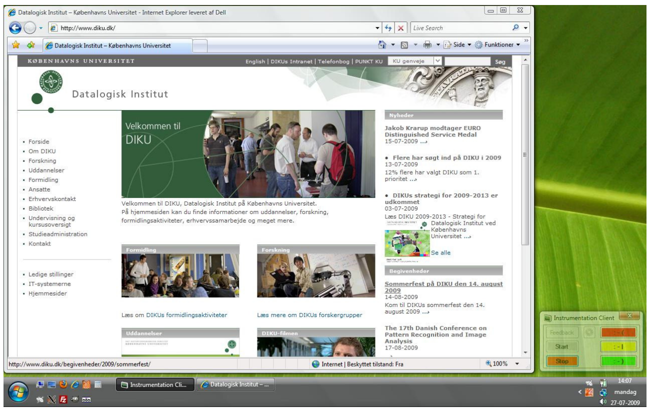


Figure 2. The small Instrumentation Client to the right of a web browser. Transparency enables placement in front of the application, if wanted.

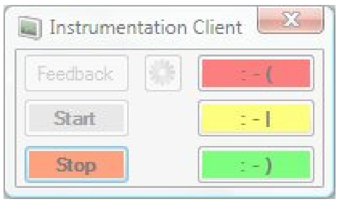


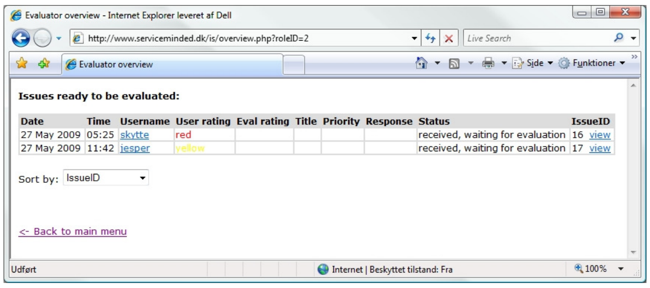
Figure 3. Close-up of the Instrumentation Client. The user's severity rating of a usability issue is recorded by pressing one of the three colored buttons to the right.

After pressing the button, the user has to think out loud and explain the experienced issue vocally to the microphone, and maybe demonstrate further on screen by using the application if the issue was not evident from what was recorded before the button press.

A timestamp at the user's button activation is collected in the Instrumentation Client and is later used to create a bookmark in the recorded video of the user's screen activity and voice commentary.

The Evaluator Role

When data is received on the Instrumentation Server the evaluator can log into the Instrumentation Server web interface from anywhere through a web browser, and review new issues (see Figure 4).



Evaluator overview - Internet Explorer leveret af Dell

http://www.serviceminded.dk/is/overview.php?roleID=2

Issues ready to be evaluated:

Date	Time	Username	User rating	Eval rating	Title	Priority	Response	Status	IssueID
27 May 2009	05:25	skytte	red					received, waiting for evaluation	16 view
27 May 2009	11:42	jesper	yellow					received, waiting for evaluation	17 view

Sort by: IssueID

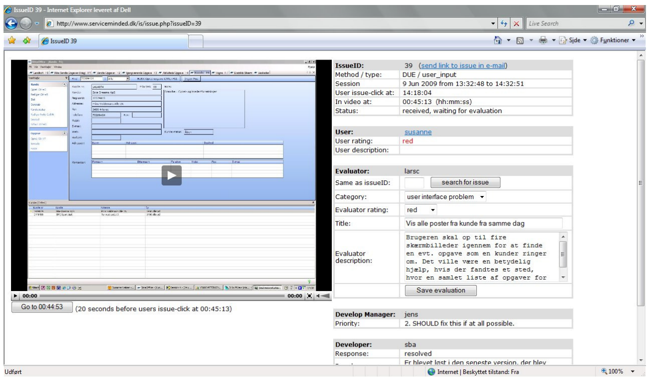
<- Back to main menu

Udført — Internet | Beskyttet tilstand: Fra — 100%

Figure 4. Two new issues ready for evaluation on the Instrumentation Server

The Instrumentation Server's web interface ensures that the evaluator can run through all new issues in an efficient workflow.

For each issue the evaluator can jump to where the user experienced an issue—20 seconds before the user pressed the button—review what happened in the video of the user's screen, hear the user's explanation, and see the user's severity rating (see Figure 5).



IssueID 39 - Internet Explorer leveret af Dell

http://www.serviceminded.dk/is/issue.php?issueID=39

Go to 00:44:53 (20 seconds before users issue-click at 00:45:13)

IssueID:	39 (send link to issue in e-mail)
Method / type:	DUE / user_input
Session	9 Jun 2009 from 13:32:48 to 14:32:51
User issue-click at:	14:18:04
In video at:	00:45:13 (hh:mm:ss)
Status:	received, waiting for evaluation
User:	susanne
User rating:	red
User description:	
Evaluator:	larsc
Same as issueID:	search for issue
Category:	user interface problem
Evaluator rating:	red
Title:	Vis alle poster fra kunde fra samme dag
Evaluator description:	Brugeren skal op til fire skærmbilleder igennem for at finde en evt. opgave som en kunder ringer om. Det ville være en betydelig hjælp, hvis der fandtes et sted, hvor en samlet liste af opgaver for
	Save evaluation
Develop Manager:	jens
Priority:	2. SHOULD fix this if at all possible.
Developer:	sba
Response:	resolved
	Er blevet løst i den seneste version, der blev

Udført — Internet | Beskyttet tilstand: Fra — 100%

Figure 5. Issue evaluation on the Instrumentation Server. User screen-video running to the left, evaluations written to the right. Full-screen viewing of video is possible.

The evaluator then decides whether the issue is a usability problem or not. If it is judged to be a problem, it is described and rated for its severity, or classified as a *same-as* problem, that is a problem found before, which then is linked to the prior description of the same problem.

The evaluators do not have to use time on analysing large amounts of instrumented data and reviewing many hours of video to identify usability problems. Instead evaluators rely on the user's ability to report when they experience something as troublesome and primarily have to decide if the reported issue is a usability problem or not.

The Development Manager and Developer Role

Development manager and developers run through a similar workflow as the evaluator, prioritizing and responding to the issues that are approved as new problems by the evaluator.

Summary of the Instrumentation Framework Characteristics

This type of user-controlled instrumentation has the advantage that developers do not have to build anything into their application, and companies need not to be concerned about how the instrumentation will affect their application.

The Instrumentation Client can be run automatically on the user's computer at specified times, taking care of data collection, and letting the user work and use the application without disturbances. The user presses a single button when experiencing something as troublesome, shortly explains and demonstrates the experienced problem, and carries on work.

The Instrumentation Server automatically prepares data for evaluators and developers, making it possible for them to jump directly to the issues in the video, and concentrate on reviewing and assessing these.

Further, the issues are conveniently stored for communication and future use in the development cycle.

CASE STUDY

As this was the first practical use of DUE, an exploratory quasi-experimental case study was chosen as the research strategy [27].

Propositions

Based on the design objectives we identified a series of propositions to be investigated in the case study:

A1: DUE supports catching usability issues (effectiveness, efficiency and satisfaction) and other types of problems.

A2: DUE reduces the workload needed for usability evaluation of software for users, developers and evaluators.

A3: DUE enables both short-term and long-term evaluations of software, with novice users and expert users.

A4: DUE strengthens the validity of identified problems:

- By the natural context of usage.
- By minimizing disturbing experimental factors.
- By enabling many test users.
- By the possibility of long-term evaluation.

A5: DUE provides an information-rich communication of the usability problems among users, evaluators, and developers.

A6: DUE contributes to minimize the evaluator effect, and support a more thorough problem analysis.

Strategy for Data Analysis

Following the general case study recommendations by Yin [27], we centered the case study on research questions designed to generate evidence that could clarify the propositions.

Building a chain of evidence (Figure 6) ensured traceability all the way from evidence to final conclusions.

Evidence ↔ fact ↔ proposition ↔ conclusion

Figure 6. Chain of evidence, making it traceable how conclusions are supported by evidence.

All qualitative and quantitative evidence from the available sources were collected, see Table 1.

Evidence was triangulated to support identified facts in the way Table 2 shows. In total 58 facts were identified.

Each fact was assessed as supporting or opposing each of the propositions, see Table 3.

Alternative explanations that could challenge the propositions (rival explanations [27]) were investigated in a similar way.

On this basis our conclusions are drawn. These are described in the *Findings and Discussion* section.

The Case Study: Production Software under Active Development

To get a thorough investigation of DUE we teamed up with a software company developing and selling a fleet management and navigation product, here called F-Man.

Because of confidentiality agreements with the developing company we cannot show detailed pictures of F-Man, describe it in detail, or mention company names.

F-Man has been on the market for a couple of years. It is under constant development and has never before been evaluated for usability, but is generally by customers considered to be user-friendly.

The users of F-Man were 16 users at a transportation company, who had been using F-Man for up to 8 months before the evaluation took place. The case study involved only one physical test site with the 16 users distributed in three offices.

The role as evaluator was carried out by an external, highly experienced usability expert from industry. This evaluator has a Ph.D. degree in computer science and human computer interaction. The development manager role was carried out by the IT manager at the transportation company, and the role as developer was carried out by two developers at the developing software company.

The case study ran over one month, in total 18 work days as it was a period with many holidays.

Installation and test of the Instrumentation Client on each participating user's computer was done by a researcher, as the Instrumentation Client was a prototype without automatic configuration implemented yet. After installation the user was instructed in approximately 5 minutes.

Due to very busy periods and customer service being a main concern for the transportation company, management preferred that users started the Instrumentation Client manually instead of this being done automatically. Because of this users were instructed to start the Instrumentation Client every day at different hours, or when they experienced an issue which could be reproduced. The Instrumentation Client automatically shut down after one hour. To help users in remembering this, a small gift was promised to the user being best at remembering the daily starts.

Source	Evidence
Usability issues	37 issues reported by users and evaluated by the evaluator. 30 issues prioritized by the development manager and responded to by developers. 37 issues reviewed for usability relevance by researcher.
Logged usage data	83 DUE sessions run by users, 57 hours total.
Observation	77 observations registered. 80 hours of user observation, 1 hour of evaluator and 4 hours of developers.
Timing of usage	13 user reports timed. 1 evaluator session timed. 1 developer session timed.
Interview	3 users interviewed in a total of 1:07 hours with 72 topics discussed. Evaluator interviewed for 32 minutes with 21 topics discussed. Development manager interviewed for 21 minutes with 27 topics discussed. 2 developers interviewed for 1:14 hours with 59 topics discussed.
Survey	8 users have answered the survey for users who have reported issues. 5 users have answered the survey for users who have not reported issues. 3 users chose not to answer any survey.
Documentation	1 e-mail from development manager to users.

Table 1. Collected evidence in the case study.

Fact 43 (A2, A5, A6)	**Issues are easy to understand for evaluator, development manager and developers.**	
Evidence	**Source**	**ID**
All users who have reported issues state that the evaluator has understood these correctly.	Survey	S16
Evaluator understands issue at first review of video in 11 out of 13 issues.	Observation	O35
Development manager states that it is easy to understand issues.	Interview	I100
Developers state that it is easy to understand issues.	Interview	I159
Further evidence: O40, O50, O73, I28, I77, I79, I107, I115.		

Table 2. Example illustrating how Fact 43 was triangulated by the collected evidence.

To protect user privacy only the screen where F-Man ran was recorded. Most users had two screens. Furthermore users were able to stop the Instrumentation Client at any point and start a new session later.

Login information to the Instrumentation Server was e-mailed to the evaluator who also received a short instruction by phone. The development manager and developers also received login information by e-mail and were given a 15 minutes demonstration. Their reviews of issues were done from work or at home via a web browser.

Automatic e-mail notification had not yet been implemented at the Instrumentation Server, and a researcher called the evaluator, development manager and developers when a suitable amount of new issues reported by users were ready for assessment. This happened four times during the case study.

A researcher was present at the transportation company observing half of the days. This set-up enabled analysis of potential influence by the researcher.

One session with the evaluator was carried out at an external usability laboratory under video-recorded observation, and one session with the developers was carried out at the company under video-recorded observation. At the end of the case study video-recorded interviews and online user surveys were carried out, see Table 1. The case study ran without any technical problems.

Collected data were transcribed, analysed and documented by the researchers for use as evidence and for the identification of facts.

FINDINGS AND DISCUSSION

A1: DUE supports catching usability issues and other types of problems

Users reported 37 issues through the Instrumentation Client. Of these issues 32 were usability problems, 4 problems were primarily of technical nature. A single problem could be attributed to lack of knowledge among users. The evaluator, the development manager and developers agreed that these issues were valid problems. 13 problems were considered as "same-as" by the evaluator. Thus in total 19 unique problems were caught. Of the technical problems one was a "same-as" problem, thus 3 unique technical problems were caught.

Proposition A5		**DUE gives an information-rich communication of the usability problems among users, evaluators, and developers**	
Support	**Oppose**	**Fact**	**Description**
X		30	Structure and documentation of issues is beneficial.
X		37	Real problem can be hidden for the user, but developers easily recognize the problem.
X		43	Issues are easy to understand for evaluator, development manager and developers.
	X	9	DUE is supplemented or overridden if serious problem prevents work at all.
	X	15	Some users find reporting to the microphone unpleasant.
-	-	-	etc.

Table 3. Example of proposition with supporting and opposing facts.

Through the researcher's observations and interviews 4 problems were identified, which were not reported by the users. Three of these are considered relevant usability issues. One was cosmetic and only observed once during 80 hours of observation. A second was severe, but very close to other reports (slow search facilities). The third problem was unclearly described and occurred for one user during very busy periods. The fourth was of internal technical nature, and did not show up in the F-Man user interface.

Our analyses indicate that the identified problems primarily are problems concerning efficiency and satisfaction, while questions concerning effectiveness and general workflow issues get less attention. To get more attention to effectiveness problems it will probably be necessary to adjust the DUE technique by having users that are less habituated with the evaluated application. Other factors influence the DUE technique's ability to catch problems, for instance the periodic activation of the Instrumentation Client, and personal differences among the users such as willingness to speak to a microphone without personal contact with the receiver of the message.

Severe usability problems were uncovered with DUE, for example a seemingly simple problem with the pick-up and delivery ordering not being visible in F-Man. This had a huge impact in busy hours, as it resulted in users having to call drivers constantly over the phone. The problem resulted in poor service for customers calling to hear when they could expect pick-up or delivery, and on a daily basis many work hours were wasted. Finding such a problem was heavily dependent on the evaluation taking place in the real use context.

An important quality gained by DUE was that the developers got a very direct and sometimes completely new insight into intricacies of the users' work situation.

In summary, the DUE technique has shown its ability to capture a significant share of the problems that the users of the application perceive as important, problems that the developers and the external usability expert agree to recognize as important.

A2: DUE reduces the workload for users, developers and evaluators

The total amount of time used by users, evaluator, development manager and developers for this DUE evaluation is for instruction 2 hours 20 minutes, and 8 hours 3 minutes to locate, understand, describe and rate the 37 reported issues.

All the empirical data indicate a small workload induced on the participants in carrying out the evaluation. Test tasks are unnecessary, as a broad range of tasks follow by the user's ordinary work processes. The reporting of an issue is activated by a mouse click and includes the user's spoken explanation. The average reporting time of an issue in this experiment was 1 minute 15 seconds. The users found it easy to use the Instrumentation Client. The evaluator reported that he found the issues unusually clearly documented, and thus easy and fast to understand and assess with an average time of 2 minutes 45 seconds spent per issue. The development manager and developers had a similar experience and on average spent a similar amount of time to understand and assess the issues.

As mentioned, the transportation company who participated in the experiment wanted the Instrumentation Client to be turned off at busy periods. In these periods this caused a few seconds waiting time to start up the Instrumentation Client and report an issue. Also some of the periodical activations, which was part of the experimental setup, were forgotten by users. This indicates that automatic starting of the Instrumentation Client would have reduced the workload even more and would have been more convenient for the user.

In summary, the case study shows how DUE induces a small workload on the users, the evaluator and the developers.

A3: DUE enables both short-term and long-term evaluations, with novice users and expert users

This proposition has not been directly tested in this case study, because all the participating users were experienced daily users of F-Man. The experiment did show, however, that using DUE was easy; the experiment was running for 4 weeks without any suspension. Testing with novice users would make it possible to follow how usability problems change as users move from novice to expert. Such studies can be very useful in research of learnability of complex software, an important aspect that we know only little about [7].

A4: DUE strengthens the validity of identified problems

To investigate this proposition we are distinguishing between internal, external, and ecological validity. These types of validity can be clarified by discussing the following characteristics of the DUE technique: (1) natural context of usage—this influences ecological validity; (2) minimizing disturbing experimental factors—this influences internal and ecological validity; (3) enabling many test users—this influences external validity; (4) the possibility of long-term evaluation—this influences external validity.

NATURAL CONTEXT OF USAGE: All users were able to use DUE in the form of the Instrumentation Client at their workplace. A few users with older PC's experienced a decreased performance, but only one found this to be relatively disturbing.

The users' busyness did influence the reporting activity, but we found no evidence that this has reduced the validity of the reported issues.

Reporting activity was influenced if users heard others report what they thought to be the same issue. Thus some under-reporting of problems in this study took place, which undermines the possibility of using the frequency of a problem report as indicator of seriousness. The fact that the case study took place at just one location in a partly open office landscape made this factor important. If an evaluation can be distributed over different physical locations this factor will play a less important role.

The developers and the evaluator emphasized that the reported issues have given them an insight into the users' natural work settings, including a number of especially challenging situations, which otherwise had been difficult to achieve. This confirms that DUE has contributed important contextual information which has strengthened the ecological validity of the usability problems.

MINIMISING DISTURBING EXPERIMENTAL FACTORS: Compared to conventional think-aloud testing the evaluator found DUE recordings to be much less influenced by irrelevant and disturbing factors.

The most influencing disturbing factors seem to have been the discomfort for some users of (a) speaking to a microphone, and (b) being monitored by the Instrumentation Client. Some users say that they find it difficult to describe an issue; but in only one case a description was found to be unclear and not really understandable to the developers and the evaluator. In this case the user has afterwards explained that she—in the situation of reporting—was fiery and thus unable to explain herself.

Concerning both being monitored and speaking to a microphone, it must be noted that the majority stated that they did not feel any discomfort. For some users, however, this discomfort can have influenced their reporting activity. In the case study, the Instrumentation Client was running for one hour a day; and only one of the user's two screens was video recorded. One could go further in providing the integrity of the user's privacy, for instance by letting the user go through all of his/her video recordings, and delete clips that are undesirable.

ENABLING MANY TEST USERS: Based on measures of "time used" for users, developers, and the evaluator from the case study, we have estimated that a DUE evaluation with 84 users can be done during a period of one month with an effective work effort of one week (30 hours) by an evaluator (Table 4), a development manager, and the group of developers. The participating users will on average contribute 10 minutes each. The instruction of how to use the instrumentation client will take further 5 minutes per user. The calculation is just a rough indication as it is based upon measurements of time used from this single case study. Time spent for a DUE evaluation will be heavily dependent of the number of issues reported, which again will be dependent of the software being evaluated and the involvement of the participating users. Note, that a DUE evaluation can be scaled, even after it has been started, by connecting or disconnecting users.

In this case study the users could work productively during the evaluation. If the software is less mature and work only is carried out for evaluation purposes, time spent by the user must be fully accounted to the evaluation cost. In the described example in Table 4 this would be 1 hour per day, totaling 22 hours per user.

POSSIBILITY OF LONG-TERM EVALUATION: The case study lasted 4 weeks demonstrating that a DUE evaluation can go on for a long time. Based on the measures of time spent by the different participants in the current case study we can estimate that a DUE evaluation with 10 users lasting 8.5 months can be carried out with an effective work effort of one week by the evaluator, the development manager, and the group of developers (Table 4). Each of the 10 users can be expected to use 1 hour 21 minutes for reporting usability issues during these 8.5 months. Again, these numbers are just rough indications.

Observed factors
Evaluator, development manager, and developer's issue review takes 2:45 minutes per issue on average. User report rate is 0.35 issues per hour on average. User issue reporting takes 1:15 minutes per issue on average.

Assumptions common to example 1 and 2
Evaluator, development manager, and developers each have 30 hours available for evaluation. 30 hours / 2:45 minutes per issue ≈ 654 issues in total can be evaluated.
The Instrumentation Client runs for 1 hour each workday.
One month is 22 workdays.

Example 1: Number of users possible
(Evaluation can run for one month)
22 workdays x 1 hour x 0.35 rate = 7.7 issues per user on average.
654 issues / 7.7 issues per user ≈ ***84 users.***
7.7 issues x 1:15 minutes ≈ 9:37 minutes spent per user on reporting.

Example 2: Time span possible
(10 users are representative of the end user population)
10 users x 1 hour x 0.35 rate = 3.5 issues per workday on average.
654 issues / 3.5 issues per workday ≈ 186 workdays ≈ ***8.5 months.***
186 workdays x 1 hour x 0.35 rate x 1:15 minutes ≈ 81:23 minutes spent per user on reporting.

Table 4. Two hypothetical examples to illustrate number of users and time span possible in large-scale DUE evaluations. The average figures in the assumptions are based on measures from the case study.

A5: DUE provides an information-rich communication among users, evaluators, and developers

For better communication, it is important that the usability problems are described as accurately as possible along with information-rich contextual information. Some of the developers told how they, before DUE, often experienced that they—after having spoken with a user about a problem on the phone—had not really understood the problem. The development manager added that he and the developers also experienced difficulties in their communication about usability problems.

By collecting usability issues at one easily accessible place, the Instrumentation Server, the participants had one common information repository to service everybody. The study showed that the problems were easily understood by developers and the evaluator. They emphasized the effective way of combining video clips, and the user's spoken explanations along with their mouse pointing. Often these descriptions of the issues contained contextual information, that were valuable in understanding the importance of an issue, for instance situations with interruptions by telephone calls, and difficulties in taking up work after being interrupted.

As mentioned above, some participants were not comfortable talking to the microphone or being video recorded during work; this can limit the usage of DUE, see the paragraph *Minimizing disturbing experimental factors*.

A6: DUE contributes to minimize the evaluator effect, and support a more thorough problem analysis

The detection of problems is entirely left to the users, and therefore the contribution to the evaluator effect from the evaluator’s problem detection activity is avoided. The rest of the evaluator effect, that is: (1) the evaluator’s severity rating of usability problems, and (2) the matching of similar reported usability issues as "same-as", is still a challenge. Instead we are left with a "user effect", in the sense that users identify different sets of usability problems when exposed to the same test setting [21]. With DUE we expect to meet this by involving a higher number of users, in a larger time span, than would be possible in conventional usability testing based on think-aloud protocols.

In the current study we have observed a tendency of the evaluator to rate the problems as more serious than the developers and the users. The evaluator, who was invited as an external usability expert, did only have a very brief introduction to the application domain and the users workflow. It could have been interesting to involve evaluators with a solid domain insight [6].

The task of matching usability problems as "same as" is an important aspect of usability analysis and far from an easily solved matter [13]. The repository of usability problems can to some extent support such analyses, for instance by grouping problems according to where in the application they were reported; also annotations to the problems might be to some use. But such use of the repository has not been experimented in this study.

The evaluator found it easy to use the Instrumentation Server and emphasized that the issues were easy to understand—with one exception, the user reporting while fiery as mentioned above. The evaluator found it especially handy that he usually did not have to forward or rewind in the video. Starting the video clip 20 seconds before the user actually clicked for reporting an issue, established a sufficient understanding of the user’s situation. The 20 seconds worked fine and matches the “specious present” described by William James [18]. The evaluator found that the DUE evaluation seemed to have even more practical value than think-aloud testing. The provided information was vivid, informative, and convenient to use, he said.

CONCLUSION

Usability evaluation draws heavily on scarce resources such as expertise in interaction design, programming, insight into the application domain and the company’s needs. Further, usability evaluation is time consuming and complex in many ways.

In an effort to show that these difficulties might be reduced without compromising the quality of the evaluation work, a technique called Distributed Usability Evaluation (DUE) was designed. DUE is based on a tool with user-controlled problem detection, automation and non-intrusive instrumentation with capture of screen video and voice recordings. The tool can be used for evaluation of prototypes and running versions of application software.

In a case study in industry DUE was demonstrated to be effective for catching important usability problems in a complex software product. The usability issues found related mainly to efficiency and satisfaction aspects, while issues related to overall effectiveness were rarely discovered. This indicates that DUE evaluations should be combined with evaluation techniques directed at identifying effectiveness issues. The contribution to the evaluator effect from the problem detection activity was partly eliminated by letting the users detect the problems.

The involved stakeholders: users, developers, a development manager, and a usability expert, all expressed a high degree of satisfaction with the DUE technique. Information-rich descriptions of usability findings were stored in a common repository, which was easily accessible by all stakeholders. This served to clarify the understanding of problems and made the communication about possible solutions more concrete, avoiding many misunderstandings. The developers emphasized the usefulness of the direct and sometimes completely new insight into intricacies of the users’ work situation.

Overall, the stakeholders assessed the quality and the validity of the DUE evaluation to be high. The important factors were the natural use context, few disturbing experimental factors, the ability of engaging many test users, and running the evaluation for a long period of time. Some users, however, reported discomfort by speaking to a microphone and by being monitored while working.

The small workload to run a DUE evaluation makes it practically possible to perform large-scale usability evaluations, potentially revealing usability problems related to learnability as the user goes from novice to expert. Further, investigating the usability of a broad range of functionality in large and complex software becomes possible.

ACKNOWLEDGMENTS

We would like to thank Dina Friis, Guðrún Hulda Jónsdóttir Johannessen, Esben Warming Pedersen, Kasper Hornbæk, Claus Topsøe-Jensen, Mikkel R. Jakobsen and Jakob G. Simonsen for valuable comments to an earlier version of this paper. Also anonymous reviewers gave many insightful comments. Special thanks go to Tobias Uldall-Espersen, the transportation company, and the software developing company—without their insight and generous cooperation this study had been impossible.

REFERENCES

1. Bateman, S., Gutwin, C., Osgood, N., and McCalla, G. 2009. Interactive usability instrumentation. In *Proceedings of the 1st ACM SIGCHI Symposium on Engineering interactive Computing Systems*, 45-54.
2. Castillo, J. C., Hartson, H. R., and Hix, D. 1998. Remote usability evaluation: can users report their own critical incidents?. In *CHI 98 Conference Summary on Human Factors in Computing Systems*, 253-254.
3. Coolican, H., *Research Methods and Statistics in Psychology*, 3rd ed. Hodder & Stoughton, 1999.
4. Frøkjær, E., Hertzum, M., and Hornbæk, K. (2000). Measuring usability: are effectiveness, efficiency, and satisfaction really correlated? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, *CHI Letters*, 2(1), 345-352.
5. Frøkjær, E. & Hornbæk, K. Cooperative Usability Testing: Complementing Usability Tests With User-Supported Interpretation Sessions, *Proc. CHI '05 extended abstracts on Human factors in computing systems*, ACM Press (2005), 1383-1386.
6. Følstad, A. Work-Domaine Experts As Evaluators: Usability Inspection of Domain-Specific Work-Support Systems, *IJHCI*, 22, 3 (2007), 217-245.
7. Grossman, T., Fitzmaurice, G., and Attar, R. 2009. A survey of software learnability: metrics, methodologies and guidelines. In *Proceedings of the 27th international Conference on Human Factors in Computing Systems*, 649-658.
8. Hartson, H. R., Castillo, J. C., Kelso, J., and Neale, W. C. 1996. Remote evaluation: the network as an extension of the usability laboratory. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 228-235.
9. Hartson, H. R. and Castillo, J. C. 1998. Remote evaluation for post-deployment usability improvement. In *Proceedings of the Working Conference on Advanced Visual interfaces* (L'Aquila, Italy, May 24 - 27, 1998). T. Catarci, M. F. Costabile, G. Santucci, and L. Taranfino, Eds. AVI '98. ACM, New York, NY, 22-29.
10. Hornbæk, K. 2006. Current practice in measuring usability: Challenges to usability studies and research. *Int. J. Hum.-Comput. Stud.* 64, 2 (Feb. 2006), 79-102.
11. Hornbæk, K. and Frøkjær, E. (2005). Comparing usability problems and redesign proposals as input to practical systems development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA, April 02 - 07, 2005). CHI '05. ACM, New York, NY, 391-400.
12. Hornbæk, K. and Frøkjær, E. (2008). Making use of business goals in usability evaluation: an experiment with novice evaluators. In *Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems*, 903-912.
13. Hornbæk, K. and Frøkjær, E. (2008). Comparison of techniques for matching of usability problem descriptions. *Interact. Comput.* 20, 6 (Dec. 2008), 505-514.
14. Howarth, J., Smith-Jackson, T., and Hartson, R. 2009. Supporting novice usability practitioners with usability engineering tools. *Int. J. Hum.-Comput. Stud.* 67, 6 (Jun. 2009), 533-549.
15. ISO 9241-11. *Ergonomic requirements for office work with visual display terminals (VDTs)* – Part 11: Guidance on usability. International Organization for Standardization, 1998.
16. Ivory, M. Y. and Hearst, M. A. 2001. The state of the art in automating usability evaluation of user interfaces. *ACM Comput. Surv.* 33, 4 (Dec. 2001), 470-516.
17. Jacobsen, N. E., Hertzum, M., & John, B. E. (1998b). The Evaluator Effect in Usability Tests. *Conference Summary of CHI'98 Conference on Human Factors in Computing Systems*, 255-256.
18. James, W. 1890. *Principles of Psychology*. Henry Holt & Co., (Chapter XV, The Perception of Time)
19. Kim, J. H., Gunn, D. V., Schuh, E., Phillips, B., Pagulayan, R. J., and Wixon, D. 2008. Tracking real-time user experience (TRUE): a comprehensive instrumentation solution for complex systems. In *Proceeding of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems*, 443-452.
20. Kjeldskov, J., Skov, M. and Stage, J. Instant Data Analysis. In *Proc. Nordichi 2004* (2004), 233-240.
21. Lewis, J. Sample size for usability studies: Additional considerations. *Human Factors, 36* (1994), 368-378.
22. Nørgaard, M. & Hornbæk, K. What Do Usability Evaluators Do in Practice?: an Explorative Study of Think-Aloud Testing, *Proc. of the 6th conference on Designing Interactive systems*, ACM (2006), 209-218.
23. Nørgaard, M. & Hornbæk, K. Working together to Improve Usability: Exploring Challenges and Successful Practices. *International Journal of Technology and Human Interaction*, 6(1) (2010), 33-53.
24. Preece, J. et al. *Human-Computer Interaction*, Addison-Wesley, 1999.
25. Spool, J. and Shroeder, W. Testing web sites: five users is nowhere near enough. In *Proc. CHI 2001* (2001), 285-286.
26. www.usertesting.com. Retrieved 2010 August 8th.
27. Yin, R. K., *Case Study Research: Design and Methods,* 4th ed. Sage Publications Inc., 2009.