What Happened to Remote Usability Testing? An Empirical Study of Three Methods

Morten Sieker Andreasen[1], **Henrik Villemann Nielsen**[2], **Simon Ormholt Schrøder**[3], **Jan Stage**[2]
[1] Systematic A/S, Søren Frichs Vej 39, DK-8000 Århus C, Denmark
[2] Aalborg University, Department of Computer Science, DK-9220 Aalborg East, Denmark
[3] Danske Bank Group, Edwin Rahrs Vej 40, DK-8220 Brabrand, Denmark
morten@sieker.dk, henrik@villemann.net, ormholt@ormholt.dk, jans@cs.aau.dk

ABSTRACT

The idea of conducting usability tests remotely emerged ten years ago. Since then, it has been studied empirically, and some software organizations employ remote methods. Yet there are still few comparisons involving more than one remote method. This paper presents results from a systematic empirical comparison of three methods for remote usability testing and a conventional laboratory-based think-aloud method. The three remote methods are a remote synchronous condition, where testing is conducted in real time but the test monitor is separated spatially from the test subjects, and two remote asynchronous conditions, where the test monitor and the test subjects are separated both spatially and temporally. The results show that the remote synchronous method is virtually equivalent to the conventional method. Thereby, it has the potential to conveniently involve broader user groups in usability testing and support new development approaches. The asynchronous methods are considerably more time-consuming for the test subjects and identify fewer usability problems, yet they may still be worthwhile.

Author Keywords
Remote testing, usability testing, empirical study

ACM Classification Keywords
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *Graphical user interfaces (GUI), Theory and methods.* D.2.2 [**Software Engineering**]: Design Tools and Techniques – *User interfaces*, *Object-oriented design methods.*

INTRODUCTION

Remote usability testing denotes a situation where "the evaluators are separated in space and/or time from users" [8]. The first methods for remote usability testing emerged about ten years ago. At that time, some empirical studies were conducted [17].

Today, we are employing new development approaches that radically increase the potential of remote usability testing. Development of Open Source Software (OSS) is very relevant for remote usability testing. The usability problems in many OSS user interfaces are well documented [6, 12, 13, 26, 29]. Moreover, a recent study of OSS developers showed that there is a genuine interest from OSS developers in improving the usability of OSS [2]. OSS development is characterized by distributed collaboration between contributors to a specific project. A project can have hundreds of contributors spread worldwide [10, 24]. This makes it hard to employ conventional usability testing methods.

Outsourcing and global software development are other contemporary development approaches where remote usability testing is very relevant. With both approaches, developers, evaluators and users are distributed across organizations and time zones. These circumstances make conventional usability testing considerably more complex and challenging [25].

Changes in software development such as these new approaches imply that remote usability testing is becoming increasingly important as an alternative to conventional usability testing. Separating evaluators and users in time and space makes it convenient to involve broader user groups in usability testing across organizational and geographical boundaries. In order to realize this potential, we need systematic comparisons of remote usability testing methods. This will help researchers and practitioners understand options for and tradeoffs between the methods.

This paper provides results from a systematic empirical comparison of methods for remote usability testing. In the following section, we describe previous research on remote usability testing. This survey forms the basis for selecting the three remote methods that we have compared. Next, we present the experimental method of our empirical study. Then we present the results of the study. This is followed by a discussion of implications of the results of the study. Finally, we provide the conclusion.

CHI 2007, April 28–May 3, 2007, San Jose, California, USA.
Copyright 2007 ACM 978-1-59593-593-9/07/0004...$5.00.

	Synchronous methods		Asynchronous methods		
	Usability evaluation	Usability inspection	Self administered web study	Self reporting of critical incident	Logged use pattern
Text communication	[3, 4, 5]	[22]	[28]	[16, 8]	
Questionnaire or multiple choice	[3, 4, 5, 19, 17, 34, 1]	[22]	[27]	[17]	[35, 32]
Post-test interview			[28]		
Workflow logging	[17]	[22]			[35, 32]
Screen shot (still image)	[15]	[22]			
Live observation	[31, 14, 3, 4, 5]				
Audio communication	[31, 14, 3, 4, 5, 19, 7, 17, 23, 34, 15, 1, 11]	[9]		[17]	
Video capture of screen	[31, 14, 3, 4, 5, 19, 7, 17, 23, 34, 15, 1, 11]	[9]		[16, 8, 17]	
Video capture of face	[19, 15, 11]				

Table 1. Remote usability testing methods.

RELATED WORK

We have conducted a systematic survey of literature on methods for remote usability testing. We based the survey on a definition of remote usability testing as a situation where "the evaluators are separated in space and/or time from users" [8]. This definition emphasizes two general types of methods; synchronous and asynchronous. With a synchronous method, the evaluator is separated from the user spatially, but not temporally. When conducting an asynchronous test, the evaluator is separated from the user both temporally and spatially.

Based on the literature about remote usability testing, we identified five different methods. Two of them are synchronous and three are asynchronous. We have related the literature on remote usability testing to these five methods, see Table 1.

Most synchronous methods conduct remote usability testing by simulating a conventional laboratory-based think-aloud test. This is achieved by using video and audio connections combined with remote desktop sharing [11, 15, 19]. The advantages found include cost efficiency, a more diverse pool of suitable test users, and identification of the same number of problems as a conventional usability test; and in some cases even more. The disadvantages are problems in building up trust between test monitor and user, a longer setup time, and severe difficulties in order to re-establish the test setup if there is a malfunction in the hardware or software [11].

The other group of synchronous usability testing methods is based on inspection. Two references describe experiments where synchronous remote usability inspection was performed either as collaboration between usability experts that inspects a system together [9] or as a walkthrough conducted by geographically dispersed members of a development team [22].

These results illustrate that there is a substantial body of literature that provides guidance on synchronous methods. Most of the references concentrate on giving pros and cons as well as practical advice with a specific method. In many cases, there is only little description of the experimental method and the data analysis of the underlying study, and often the empirical data are not provided. However, there are some notable exceptions where papers report from empirical comparisons [7, 23, 34], usually between a single remote method and the conventional method. These references will be discussed in detail in the Discussion section below.

The literature on asynchronous testing methods is more limited. Two references discuss strengths and weaknesses of asynchronous remote testing based on case studies [17, 28]. Another reference reports from a self-administered web study, where the user filled out a questionnaire during the test. This is compared to a conventional usability test in a laboratory. This study revealed a number of disadvantages with remote testing as the frequency of completion amongst the users was low, it was very time consuming, and it provided less qualitative information [27].

A different approach is to make the users themselves identify and report the critical incidents they experience. In a study of this method, the users were taught how to do this with minimal training. The study showed that the users only missed few of the critical issues found through conventional testing. The method also proved to be both cost and labor efficient as much of the work was moved from the evaluators to the users [8, 16, 17]. Thus with this method, the users are performing much of the evaluation work themselves with only a minimal involvement of expert evaluators.

Finally, it has been studied how automatic logging of use patterns employed by the users can help identify usability problems [32, 35]. This study found the same disadvantages as Olmsted *et al.* [27] when analyzing logged use patterns. The lack of accurate qualitative data made the analysis difficult and it proved to be a lot less efficient compared to conventional usability testing methods [35].

This survey illustrates that the empirical studies of remote usability testing methods are generally characterized by studying only a single remote method or comparing one such method to the conventional method with laboratory-based think-aloud testing (sometimes also referred to as local usability testing).

METHOD

We have conducted a systematic experimental comparison of three methods for remote usability testing and a conventional laboratory-based think-aloud method. The three remote methods were selected from the five methods discussed above. The conventional method was used as a benchmark. Thus we compared the following four methods:

- Laboratory testing (LAB)
- Remote synchronous testing (RS)
- Remote asynchronous expert testing (AE)
- Remote asynchronous user testing (AU)

Laboratory testing (LAB) is the conventional usability testing method. Remote synchronous testing (RS) is conducted in real time but the test monitor is separated spatially from the test subjects. Remote asynchronous expert testing (AE) is conducted with the test monitor and the test subjects separated both spatially and temporally. The test subjects are usability experts. Remote asynchronous user testing is conducted in the same way but with ordinary users as test subjects.

In the rest of this section, we describe how we studied and compared these four methods. First, we describe the factors that were common for all four conditions. Second, we describe the specific aspects of each condition.

Participants: A total of 24 *test subjects*, 14 male and 10 female, participated as users/experts in the four different conditions. They were all students at Aalborg University and aged between 19 and 30 (mean=25.13, SD=3). The average ages for the tests subjects in each condition are very similar as shown in Table 2. All test subjects had experience using a computer and the Internet. The six test subjects in the AE condition had received formal training in usability evaluation through their education. The other 18 test subjects had not received any formal training in usability evaluation. They were randomly assigned as test subjects to one of the three other conditions (LAB, RS and AE). The test subjects received snacks and beverages as compensation for their participation.

	Number of Test Subjects		
	Female	**Male**	**Sum**
LAB	4 (26.3)	2 (21.5)	6 (24.7)
RS	2 (26.5)	4 (24.0)	6 (24.8)
AE	2 (26.5)	4 (26.8)	6 (26.0)
AU	2 (23.0)	4 (26.0)	6 (25.0)

Table 2. Number of test subjects in the four conditions. The number in parenthesis denotes the average age.

One of the authors of this paper served as *test monitor* (moderator) in all twelve synchronous tests (LAB and RS). Three of the authors served as *evaluators* by carrying out the data analysis that is described below.

System: We tested the email client Mozilla Thunderbird 1.5. We wanted to test a system within a domain that was familiar to the test subjects. During the screening and selection of test subjects we made sure that none of them had experience with Thunderbird; but they had all used an e-mail client like Outlook or Netscape mail, so they were familiar with the basic concepts of an e-mail application.

Tasks: We developed nine tasks that the test subjects should complete during the tests. They are shown in Table 3. These tasks were used in all four conditions.

Task #	**Description**
1	Create a new email account (data provided)
2	Check the number of new emails in the inbox of this account
3	Create a folder with a name (provided) and make a mail filter that automatically moves emails that has the folder name in the subject line into this folder
4	Run the mail filter just made on the emails that were in the inbox and determine the number of emails in the folder
5	Create a contact (data provided)
6	Create a contact based on an email received from a person (name provided)
7	Activate the spam filter (settings provided)
8	Find suspicious emails in the inbox, mark them as spam and check if they were automatically deleted
9	Find an email in the inbox (specified by subject line contents), mark it with a label (provided) and note what happened

Table 3. The tasks used in the usability tests.

Laboratory Testing (LAB)

Setting: The LAB tests were based on the conventional think-aloud protocol [30] and were conducted in a state-of-the-art usability laboratory. The test subject and the test monitor performed the tests in a designated test room fitted with video cameras and microphones (Figure 1, room A). The test monitor and test subject were both seated in front of the PC. The role of the test monitor was primarily to ensure that the test subjects were thinking aloud while performing tasks, but he would also ask test subjects to proceed to the next task if they got stuck.

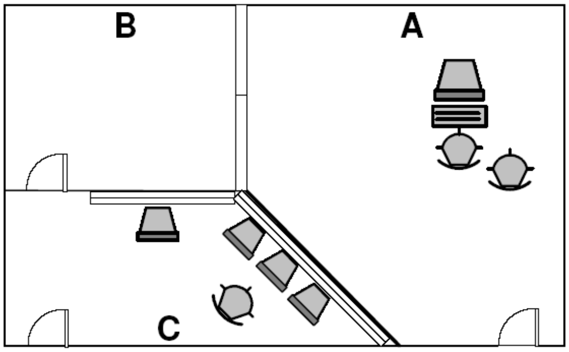

Figure 1. The setting for the LAB test.

Procedure: The test monitor introduced the test subjects to 'thinking aloud'. Then the test subjects were asked to solve the nine tasks. We did not specify a time limit, but encouraged the test subjects to try to solve all tasks without help from the test monitor. After the test session, the test subject was debriefed and interviewed about the test method.

Data collection: We recorded both audio and video feeds. The video feed consisted of the test subject's desktop and a small video image of the test subject's face in the bottom right hand corner of the screen.

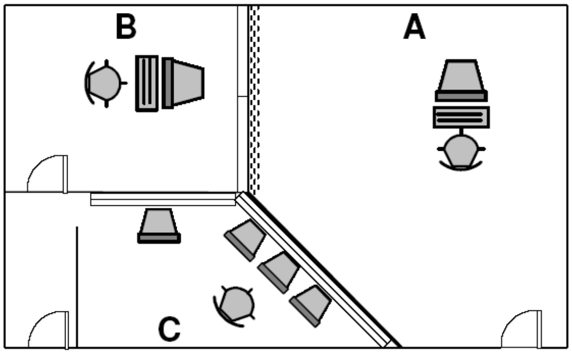

Figure 2. The setting for the RS test.

Remote Synchronous Testing (RS)

Setting: The RS tests were based on the literature described in Related Work, where a simulated laboratory test using web-cams, remote desktop connections and audio was found to be the most frequently used and effective setup. We performed these tests in the usability laboratory, with a setup that simulated such a remote testing environment. The test monitor and test subject were in separate rooms (Figure 2, room A (test monitor) and room B (test subject)), and they could only communicate through web-cams and audio connection (Figure 3). We chose VNC (Virtual Network Computing) and Microsoft Netmeeting as the testing platform since this software allowed shared desktop and communication via web-cams. We selected Skype for the audio communication, since the audio quality in Netmeeting was not satisfactory.

Procedure: The procedure in the RS tests was the same as in the LAB tests.

Data collection: We recorded the audio and video that the test monitor experienced through VNC, web-cam and Skype. The video feed consisted of the view of the test subject's desktop as provided by Netmeeting and a small video image from the test subject's web-cam in the lower right hand corner. During the test, the test subject had a web-cam image of the test monitor in the lower right hand corner of the screen, but this was not visible in the recorded video.

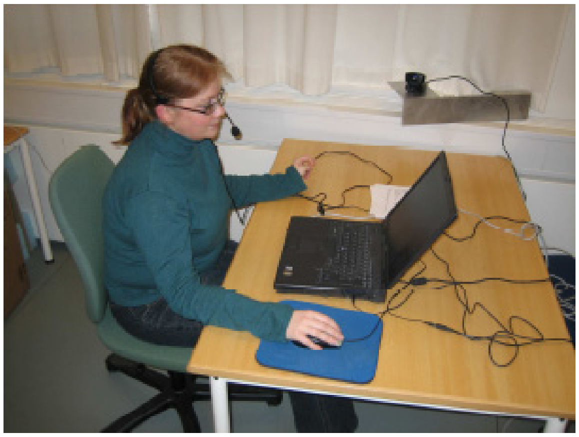

Figure 3. A test subject in the RS test.

Remote Asynchronous Testing (AE and AU)

Setting: The method used in both of the asynchronous tests was inspired by particularly one remote method where the users themselves report the 'critical incidents' they experience [8, 16, 17]. With this method, the test subjects not only perform the tests but also identify usability problems in the software that is tested. This has the advantage of relieving the evaluators of a considerable workload. In these tests, we wanted to examine if users without any formal usability knowledge were able to generate results that were useful for a usability evaluation. To better understand whether this was possible, we conducted the tests in two conditions with two different groups of test subjects: usability experts (AE) and ordinary users (AU). Both conditions were conducted in a remote location at the test subjects' own computers at a time that was convenient for them. Their task solving process was guided by an online questionnaire that the tests subjects would read and answer.

Procedure: We made an installation manual to Thunderbird and included that as the first page of the online questionnaire. This was done in order to minimize the contact between test subjects and evaluators before and during the tests. The tasks were also included as an integrated part of the online questionnaire. The test subject would first install the system. Then they would work through the online questionnaire task by task, and for each task they could report any identified usability problems into the questionnaire.

Data collection: The data collection for the two asynchronous conditions was done solely through the online questionnaire that was constructed in UCCASS (Unit Command Climate Assessment and Survey System). It gathered the input from the test subjects and stored it in a MySQL database. The questionnaire enabled the test subjects to categorize the identified problems as ‘small’, ‘medium’ and ‘large’. These categories were correlated to the commonly used classifications ‘cosmetic’, ‘serious’, and ‘critical’. The test subjects were presented with a table specifying how to classify a specific usability problem, see Table 4. Furthermore, they were asked to log the location in the program where they encountered a problem and describe how it influenced the completion of the task.

Problem Severity	Delay in task completion	Irritation	Expectation to system behaviour
Small	Less than 30 seconds delay	Slight irritation	Minor difference in expected action
Medium	More than 30 seconds delay	Average irritation	Significant difference in expected action
Large	Could not complete the task	High irritation	Critical difference in expected action

Table 4. Guidelines used for categorizing the severity of usability problems.

Data Analysis

The data analysis was conducted by three of the authors of this paper. The analysis procedure was designed carefully to maximize inter-evaluator reliability and to minimize the subjective bias of the evaluators [21], given that we were only three usability evaluators to condense the problem lists from the empirical data.

The data analysis was not started until all tests in all four conditions had been conducted and all data was collected. The tests produced 24 sets of data for analysis, i.e. twelve video recordings from the synchronous usability tests and twelve questionnaire responses from the asynchronous usability tests. The 24 sets of data were given a random identifier to avoid that an evaluator would know which data set he was analyzing. Then each evaluator randomly drew the order in which he would analyze the 24 sets of data.

The three evaluators conducted the whole data analysis independently of each other. Each evaluator analyzed one data set at a time. In each set of data, he identified usability problems and numbered them with a unique identifier to make it possible to trace each problem back to the original occurrence. Each problem was also categorized as critical, serious or cosmetic. The evaluators also made their own categorization of the usability problems from the asynchronous tests, independently of the categorizations made by the test subjects. Half of the data sets were video recordings that were analyzed systematically to identify usability problems experienced by the test subject, while the other half were questionnaire responses that were processed to compile a list of the usability problems reported by the test subject. Through this analysis each evaluator generated his own list of usability problem for the tested software. This process took approximately 42 hours per evaluator, a total of 126 person-hours.

When the three evaluators had completed their own problem lists, they negotiated a joint list of all identified usability problems for the tested system. This negotiation was conducted until consensus was reached. In the joint problem list, the categorization of each usability problem was determined by using a ‘worst case’ schema between the individual categorizations of the three evaluators. Thus a problem was categorized as critical if just one evaluator had categorized it as such. This process took approximately 30 person-hours.

In order to examine the reliability of the joint problem list, we calculated the evaluator effect. The evaluator effect denotes the fact that different evaluators find different usability problems. The measure determines to what extent the evaluators have found the same problems in their individual analysis [18].

$$Avg.of \quad \frac{|P_i \cap P_j|}{|P_i \cup P_j|} \quad over\,all \quad \frac{1}{2}n(n-1) \quad pairs\,of\,evaluators \quad (1)$$

In Equation 1, P_i and P_j are the problems detected by evaluator i and j and n is the number of evaluators. We calculated the evaluator effect using this any-two agreement equation in order to see how often evaluators agreed on identified usability problems. The average percentage of any-two agreement in our data analysis was 66.9 percent, see Table 5.

	E1 E2	E1 E3	E2 E3	Avg.
Problems agreed on	29	30	28	29
Number of problems	42	45	43	43.3
Any-two agreement	69.0%	66.7%	65.9%	66.9%

Table 5. Calculation of the evaluator effect between E1, E2 and E3 using the any-two agreement formula (Equation 1).

Hertzum and Jakobsen found that the average agreement between any two evaluators in twelve studies varied from 5 percent to 65 percent (avg. 22.4%, SD=19.8) [18]. Compared to these figures we achieved a very high any-two agreement measure which indicates a high level of reliability of the joint problem list.

	LAB N=6		**RS** N=6		**AE** N=6		**AU** N=6	
Task completion time: ***Average (SD)***	22:10 (05:20)		22:30 (03:31)		45:29 (18:51)		1:03:48 (48:37)	
Usability problems	#	%	#	%	#	%	#	%
Critical (24)	22	92%	22	92%	15	63%	11	46%
Serious (10)	5	50%	8	80%	3	30%	2	20%
Cosmetic (12)	8	67%	8	67%	3	25%	0	0%
Total (46)	35	76%	38	83%	21	46%	13	28%

Table 6. Task completion time (average and standard deviation) and number of identified usability problems (absolute and percentage of total number).

RESULTS

In this section, the results of our empirical study are presented.

Task Completion

We have determined the number of tasks completed by each test subject. The 24 test subjects completed an average of 8.1 out of 9 tasks. The AE and AU conditions had a higher number of solved tasks than the LAB and RS conditions, see Table 7. However, an ANOVA test gave no significant difference in the number of tasks completed in the four tests (F[3,20]=0.68, p=0.575).

	Mean value	**SD**
LAB	8.0	1.1
RS	7.5	2.1
AE	8.5	0.8
AU	8.3	0.8
Total	8.1	1.3

Table 7. Number of tasks completed.

Only task 2 and 9 were completed by all test subjects. Task 6 was most difficult as five test subjects did not complete it. Task 7 and 8 together caused difficulties for six test subjects. The remaining tasks were completed by all except two or three test subjects. Task 1, 3 and 4 were difficult for several test subjects, but they were able to complete them.

Task Completion Time

We have measured the time each test subject used to complete the tasks. The results in Table 7 reflect that in the two asynchronous conditions, the test subjects spent more time on the tasks. This is especially characteristic for the test subjects in the AU condition who had a considerably longer task completion time (M=1:03:48, SD=0:48:37) compared to the test subjects in the LAB (M=00:22:10, SD=00:05:20) and the RS condition (M=00:22:30, SD=00:03:31). An ANOVA test gave a significant difference in task completion time (F[3,20]=3.514, p=0.034). Yet, we found no pairwise differences between the conditions using a Tukey's post hoc test, at a five percent significance level.

A reason for this marked difference may be that the test subjects in the asynchronous conditions took breaks. The online questionnaire used in these conditions only recorded the starting and ending time. Therefore, we do not know if the test subjects had any breaks during the test sessions, and therefore we do not know the exact time spent on the test, which is an essential element when comparing AE, AU and LAB evaluations.

Number of Usability Problems Identified

The number of usability problems identified in the tests show considerable differences, see Table 6. The 24 usability test sessions resulted in a total of 46 usability problems. Based on the guidelines in Table 4, we categorized 24 of them as critical, 10 as serious, and 12 as cosmetic. Below, we provide further results on the number of usability problems identified in each condition.

LAB

From the LAB tests, the evaluators identified 35 of the 46 usability problems. 22 of these problems were critical, 5 were serious, and 8 were cosmetic.

	LAB	**RS**	**AE**	**AU**
LAB		(p=0.6073)	p=0.0051 *	p<0.0001 ***
RS	(p=0.6073)		p=0.0004 ***	p<0.0001 ***
AE	p=0.0051 *	p=0.0004 ***		(p=0.1300)
AU	p<0.0001 ***	p<0.0001 ***	(p=0.1300)	

**Table 8. Fisher's exact test for the total number of usability problems identified in the four conditions.
(p)=not significant, *=significant,
=very significant and *=extremely significant.**

LAB vs RS

In the RS tests, the evaluators identified 38 of the 46 overall usability problems. This is slightly better but comparable to the number of problems identified in the LAB tests. According to a Fisher's exact test (see Table 8) there is no significant difference in the number of problems identified in the two conditions (p=0.6073).

A similar result was found with the critical problems. The LAB and RS tests both identified 22 of the 24 critical problems, where 4 of the 22 problems were found in only one of the two evaluations. Thus, 20 of the critical problems identified in the two conditions were the same.

In the identification of serious and cosmetic problems the LAB and RS tests gave almost equal results. The RS tests identified 8 of 10 serious problems and 8 of 12 cosmetic problems, which was slightly better than the LAB tests. In the identification of critical (p=1.000), serious (p=0.3498), and cosmetic (p=1.000) problems no significant difference was found through a Fisher's exact test.

LAB vs AE

The AE tests identified 21 of the 46 problems, compared to the LAB evaluation that identified 35. A Fisher's exact test shows a significant difference (p=0.0051) in the number of problems identified in the two conditions, see Table 8.

In the identification of critical problems, the difference between the LAB tests and the AE tests was not as distinct, since the AE tests identified 15 of 24 critical problems against the LAB tests' identification of 22 of the 24 critical problems. This shows that even though the AE tests did not find as many problems as the LAB tests, the majority of the problems identified were critical. However, a Fisher's exact test still gives a significant difference (p=0.0363) in the number of critical problems.

The AE tests identified 3 out of 10 serious and 3 out of 12 cosmetic problems. Compared to the LAB condition Fisher's exact test does not give a significant difference in the number of serious (p=0.6499) or cosmetic (p=0.0995) problems for the AE condition.

LAB vs AU

The AU tests identified 13 of the 46 overall problems. A comparison of this result to the LAB condition through a Fisher's exact test gives an extremely significant difference (p<0.0001) as shown in Table 8.

In the identification of critical problems, the difference between the two conditions is also significant (p=0.0078), since the AU tests only identified 11 of the 24 critical problems, where the LAB tests identified 22. The majority (84,6%) of the problems identified in the AU tests were critical.

The difference in the number of serious problems identified was not significant according to the Fisher's exact test (p=0.3498). The AU tests did not find any cosmetic problems, while the LAB tests identified 8 of the 12 overall cosmetic problems. With a Fisher exact test this gives a significant difference between the two tests methods (p=0.0013).

AE vs AU

A key aim was to compare the expert and user based evaluations. This is interesting because these two conditions are identical, except for the competence of the test subjects. Table 9 shows the results of a Fisher's exact test used to compare the critical, serious and cosmetic problems identified in the two asynchronous conditions. This shows that there is no significant difference in the number of problems identified in these two conditions, despite the differences in the competence of the test subjects.

	p
Overall	0.1300
Critical	0.7702
Serious	1.0000
Cosmetic	0.2174

Table 9. Fisher's exact test for the number of usability problems in each category identified in the AU and AE conditions. p=significance level.

Number of Problems Identified in a Test Session

The number of usability problems identified in each test session also varies between the four conditions. Table 10 shows the average number of usability problems identified in each of the six test sessions that were conducted in each condition. The average number is almost the same in the RS and LAB conditions, and the Tukey comparison did not yield a significant difference in the number of problems identified (p>0.05). However, we found a very significant difference when comparing the average number of problems identified in the LAB tests with the average number of problems identified by the test subjects in the AE and AU conditions (p≤0.001).

	Mean	**SD**	**Tukey comparison**
LAB	15.33	4.41	▨
RS	16.67	2.42	p>0.05
AE	4.67	2.66	p≤0.001
AU	3.17	1.72	p≤0.001

Table 10. Average number of usability problems identified in a test session. The Tukey test compares the three remote conditions with the LAB condition.

Unique Problems

Different usability testing methods may reveal unique usability problems. We define unique problems in a manner that is inspired by the identification of action areas in Karat *et al.* [20]. In our data, we analyzed the problems that were identified in one test session only, and the problems that were identified by only one evaluation method.

Problems Identified in One Test Session Only

One type of unique problems is those identified only in one test session. As shown in the Sum column of Table 11, none of the 24 critical problems, 2 of the 10 serious problems, and 6 of the 12 cosmetic problems were identified in one test session only. This emphasizes the validity of the critical problems. On the other hand, it also shows that 50% of all the cosmetic problems were only identified in one test session.

	LAB N=6	**RS** N=6	**AE** N=6	**AU** N=6	**Sum** N=24
Critical (24)	**0** (0)	**0** (1)	**0** (0)	**0** (0)	**0** (1)
Serious (10)	**1** (1)	**0** (3)	**0** (0)	**1** (1)	**2** (5)
Cosmetic (12)	**2** (2)	**2** (2)	**2** (2)	**0** (0)	**6** (6)
Total (46)	**3** (3)	**2** (6)	**2** (2)	**1** (1)	**8** (12)

Table 11. Identification of unique problems. The number in bold is unique problems identified in one test session only. The number in parenthesis is the unique problems identified by this one method only.

Problems Identified by One Test Method Only

Another type of unique problems is those identified by only one test method. In the Sum column of Table 10 we see that 1 of the 24 critical problems, 5 of the 10 serious problems, and 6 of the 12 cosmetic problems were identified by only one of the four methods. Thus 23 of the total of 24 critical problems were identified with more than one test method. Furthermore, 50% of the serious and cosmetic problems were only identified with one test method.

Table 6 showed that the AE and AU tests identified only 63% and 46% of all critical problems, whereas the LAB and RS tests both identified 92%. This is a clear weakness for the asynchronous methods. On the other hand, Table 11 shows that the asynchronous methods did not identify unique critical problems. This shows that the problems identified with the AE or AU method were also identified with at least one of the other methods. In comparison, we found that the RS tests identified 1 critical, 3 serious, and 2 cosmetic problems that were not identified in any of the other tests. This is the largest amount of unique problems identified by any of the four methods.

Evaluators vs Self-Reporting Test Subjects

The basic idea of the two asynchronous methods is that the test subjects themselves identify and categorize usability problems. With respect to identification, we have already shown that the asynchronous methods (AE and AU) did not identify as many problems as the LAB and RS methods. If we look at the individual evaluator and test subject, we see the same difference. Based on the synchronous tests, each evaluator identified on average 4.17 problems per test session. Each test subject in the asynchronous tests, who reported his/her own problems, identified on average 1.92 problems. Thus the evaluators identified more than twice as many problems from the synchronous tests as the test subjects in the asynchronous conditions. Moreover, the most remarkable result is that the experts did not identify more problems than the ordinary users.

With respect to categorization, the difference was even more outstanding. The test subjects in the two asynchronous conditions were also supposed to categorize the problems they identified. In doing so, they should use the guidelines in Table 4. When we started analyzing the data, it quickly became evident that the categorizations made by the test subjects were not useful at all. The test subjects categorized almost all problems as cosmetic. Only 3 problems in total were deemed by them to be critical. A possible explanation may be that these test subjects solved most of the tasks (see Table 7). This may have affected their categorization, as they felt successful after managing to solve the tasks even when encountering problems. This is supported by Hartson *et al*. [17].

DISCUSSION

In our survey of related work, we found prior research on remote usability testing that compare and assess remote methods. In this section, we compare that to our results.

An early study concluded that remote usability testing in the RS condition is feasible compared to the conventional laboratory-based method (the LAB condition) [17]. The number of usability problems identified in the two conditions was very similar. These results were promising for the remote method, although the study did not allow for definite conclusions. They also assessed the AU method and concluded that the low cost for the evaluators made this method feasible. This latter conclusion was not based on a comparison. The results provided in this study are clearly in line with our findings.

A more recent study aimed to define effective tools and methods for remote usability testing [34]. Their study compared the LAB and RS conditions. They found that the remote test subjects took longer to complete most tasks which differ from our results. The remote users also made more errors. They also compared usability problems found in the two conditions. They made no statistical test, but the numbers of problems found are quite similar which is in line with our conclusion.

There is also a systematic overview of remote usability testing methods [23]. This is accompanied with results from three case studies. The first case study was of an email application, and it compared the LAB and RS conditions. They found only one significant difference between the two conditions, and they conclude that this is likely to be caused by a lack of balance across conditions. The second case study was with the same conditions, and it only gave a significant difference on task completion time. This is similar to results that others have obtained [34]. There is also a third case study, but it does not involve any comparisons. It is concluded that the LAB and RS methods produce equivalent results on usability problems. Again, this is entirely in line with our findings.

Finally, there is a recent study with a comparison of the LAB and RS conditions and with a setup that is close to ours [7]. They found no significant differences between the two conditions in terms of the number, types and severities of usability problems. They note that this is consistent with others [17] and clearly in line with our results.

The amount of comparisons involving asynchronous methods is much more limited. The original presentations of the idea of having users reporting critical incidents provide some positive experiences, but the studies are only informal [8, 16, 17]. A recent paper is based on a systematic study involving two cases where the LAB method is compared to a method that is close to the AU extended with an interview [28]. Based on the two case studies, they conclude that the results of the two conditions are comparable on some quantitative measures. Yet in one of the case studies the number of usability problems identified is more than four times higher for the LAB condition than the AU condition. This result is close to our findings.

A key difference between the LAB method and the three remote methods is that the test monitor is absent from the test situation, either spatially or both temporally and spatially. In our study, we did not experience that the 'distance' between the test monitor and test subject influenced the performance negatively. This is contrary to the suggestion of several others [1, 3, 4, 5, 11, 14, 31]. The equivalent results of the LAB and RS conditions indicate that the physical presence of the test monitor is not important. In fact, several of our test subjects in the RS condition, who had tried the LAB condition before, expressed that the RS method was less stressful. One such test subject said: "I liked this test method better than the traditional method where the test leader looks over your shoulder." Others supported this by saying that the video image of the test monitor was a positive element, since it was nice to be able to see the attitude of the moderator.

This observation is supported by another study [7]. This study compared the comfort level experienced in the LAB and RS conditions. It was concluded that the majority of participants felt that the RS condition was more convenient and would prefer this to participating in a LAB test, and nobody said the opposite.

CONCLUSION

Remote usability testing is becoming increasingly important. This paper has presented results from a systematic experimental comparison of three methods for remote usability testing and a conventional laboratory-based think-aloud test. The results show that the remote synchronous method is virtually equivalent to the conventional method. The two methods identified almost the same number of usability problems, and test subjects spent the same time completing the tasks. The former conclusion is clearly in line with most of the prior research. The results on the latter are more varied. These conclusions show that remote usability testing has the potential to cross organizational and geographical boundaries and support new approaches to software development such as outsourcing and global and open source software development.

The results on asynchronous methods are not as clear and positive. The asynchronous methods intend to move the majority of effort from expert evaluators to ordinary users. Our findings confirm that the asynchronous methods are more time-consuming for the users and identify fewer usability problems. Moreover, the tests subjects could not provide a usable categorization of the usability problems. Despite the disappointing results, these methods may still be worthwhile to use because they relieve the expert evaluators from a considerable amount of work, and enable collection of use data from a large number of participants.

The aim of this study is to increase our understanding of the tradeoffs between different types of methods for conducting usability testing. In order to move further in that direction, it would be useful with more studies of and experiments with the asynchronous methods. In addition, it would be interesting to perform comparative studies of remote usability testing methods outside the controlled environment of a usability laboratory. Finally, it would be relevant to conduct follow-up experiments with more test subjects.

ACKNOWLEDGMENTS

The research behind this paper was partly financed by the Danish Research Councils (grant number 2106-04-0022). We are very grateful to the 24 test subjects that helped us in the usability tests. We would also like to thank Mikael Skov for his help and advice regarding statistical calculations and the procedure for data analysis. Finally, we want to thank the anonymous reviewers for their comments and advice.

REFERENCES

1. Ames, M. Final Report on Remote Usability Studies. http://www.ocf.berkeley.edu/˜morganya/research/dmp/ report.html.
2. Andreasen, M. S., Nielsen, H. V., Schrøder, S. O. and Stage, J. Usability in open source software development: Opinions and practice. *Information Technology and Control 35A*, 3 (2006), 303-312.
3. Bartek, V. and Cheatham, D. Experience Remote Usability Testing, Part 1. http://www-106.ibm.com /developerworks/library/ wa-rmusts1/.
4. Bartek, V. and Cheatham, D. Experience Remote Usability Testing, Part 2. http://www-106.ibm.com/developerworks/web/ library/wa-rmusts2.html.
5. Bartek, V. and Cheatham, D. Experiences in Remote Rsability Evaluations. http://www-3.ibm.com /ibm/easy/eou ext.nsf/Publish/50 ?OpenDocument&/Publish/1116/$File/paper1116.pdf.
6. Benson, C., Muller-Prove, M. and Mzourek, J. Professional usability in open source projects: Gnome, openoffice.org, netbeans. *Proceedings of CHI 2004*, ACM Press (2004), 1083-1084.
7. Brush, A. B., Ames, M. and Davis, J. A comparison of synchronous remote and local usability studies for an

expert interface. *Proceedings of CHI 2004*, ACM Press (2004), 1179-1182.

8. Castillo, J. C., Hartson, H. R. andHix, D. Remote usability evaluation: Can users report their own critical incidents? *Proceedings of CHI 1998*, ACM Press (1998), 253-254.

9. de Vreede, G.-J., Fruhling, A. and Chakrapani, A. A repeatable collaboration process for usability testing. *Proceedings of HICSS 2005*, IEEE Computer Society (2005), Track 1, p. 46.

10. Dempsey, B. J., Weiss, D., Jones, P. and Greenberg, J. Who is an open source software developer? *Communications of the ACM 45*, 2 (2002), 67-72.

11. Dray, S. and Siegel, D. Remote possibilities?: International usability testing at a distance. *interactions 11*, 2 (2004), 10-17.

12. Eklund, S., Feldman, M., Trombley, M. and Sinha, R. Improving the Usability of Open Source Software: Usability Testing of staroffice calc. http://www.sims.berkeley.edu/˜sinha/opensource.html.

13. Frishberg, N., Dirks, A. M., Benson, C., Nickell, S. and Smith, S. Getting to know you: Open source development meets usability. *Proceedings of CHI 2002*, ACM Press (2002), 932-933.

14. Gough, D. and Phillips, H. Remote Online Usability Testing: Why, How, and When to Use it. http://www.boxesandarrows.com/view/remote online usability testing why how and when to use it.

15. Hammontree, M., Weiler, P. and Nayak, N. Remote usability testing. *Interactions 1*, 3 (1994), 21-25.

16. Hartson, H. R. and Castillo, J. C. Remote evaluation for post-deployment usability improvement. *Proceedings of AVI 1998*, ACM Press (1998), 22-29.

17. Hartson, H. R., Castillo, J. C., Kelso, J. and Neale, W. C. Remote evaluation: The network as an extension of the usability laboratory. *Proceedings of CHI 1996*, ACM Press (1996), 228-235.

18. Hertzum, M. and Jacobsen, N. E. The evaluator effect: A chilling fact about usability evaluation methods. *International Journal of Human-Computer Interaction 15*, 1 (2003), 183-204.

19. Houck-Whitaker, J. Remote Testing versus Lab Testing. http://boltpeters.com/articles/versus.html.

20. Karat, C.-M., Campbell, R. and Fiegel, T. Comparison of empirical testing and walkthrough methods in user interface evaluation. *Proceedings of CHI 1992*, ACM Press (1992), 397-404.

21. Kjeldskov, J., Skov, M. B. and Stage, J. Does time heal: A longitudinal study of usability. *Proceedings of OZCHI 2005*, ACM Press (2005), 1-10.

22. Krauss, F. S. H. Methodology for remote usability activities: A case study. *IBM Systems Journal 42*, 4 (2003), 582-593.

23. McFadden, E., Hager, D. R., Elie, C. J. and Blackwell, J. M. Remote usability evaluation: Overview and case studies. *International Journal of Human-Computer Interaction 14*, 3&4 (2002), 489-502.

24. Moon, J. Y. and Sproull, L. Essence of Distributed Work: The Case of the linux Kernel. http://www.firstmonday.org/issues/issue511/moon/index.html.

25. Murphy J., Howard S., Kjeldskov K. and Goschnick, S. Location, location, location: Challenges of outsourced usability evaluation. *Proceedings of the Workshop on Improving the Interplay between Usability Evaluation and User Interface Design, NordiCHI 200*4, Aalborg University, Department of Computer Science, HCI-Lab Report no. 2004/2 (2004), 12-15.

26. Nichols, D. M. and Twidale, M. B. *Usability and open source software*. Technical Report 10/02, Department of Computer Science, University of Waikato, Working Paper Series ISSN 1170-487X, 2002.

27. Olmsted, E. and Gill, M. In-person usability study compared with self-administered web (remote-different time-place) study: Does mode of study produce similar results? *Proceedings of UPA 2005*, UPA (2005).

28. Petrie, H., Hamilton, F., King, N. and Pavan, P. Remote usability evaluation with disabled people. *Proceedings of CHI 2006*, ACM Press (2006), 1133-1141.

29. Raymond. E. *The Revenge of the Hackers*. O'Reilly and Associates, 1999.

30. Rubin, J. *Handbook of Usability Testing*. Wiley, 1994.

31. Safire, M. Remote moderated usability. http://www.upassoc.org/usability resources/conference/2004/im safire.html, 2004.

32. Scholtz, J. Adaption of traditional usability testing methods for remote testing. *Proceedings of HICCS '01*, IEEE (2001).

33. Skov, M. B. and Stage, J. Supporting problem identification in usability evaluations. *Proceedings of OzCHI 2005*, ACM Press (2005), 1-9.

34. Thompson, K. E., Rozanski, E. P. and Haake, A. R. Here, there, anywhere: Remote usability testing that works. *Proceedings of CITC5 2004*, ACM Press (2004), 132-137.

35. Winckler, M. A. A., Freitas, C. M. D. S. and de Lima, J. V. Usability remote evaluation for www. *Proceedings of CHI 2000*, ACM Press (2000), 131-132.