The Effect of Task Assignments and Instruction Types on Remote Asynchronous Usability Testing

Anders Bruun
Aalborg University
Department of Computer Science
DK-9220 Aalborg East, Denmark
bruun@cs.aau.dk

Jan Stage
Aalborg University
Department of Computer Science
DK-9220 Aalborg East, Denmark
jans@cs.aau.dk

ABSTRACT
Remote asynchronous usability testing involves users directly in reporting usability problems. Most studies of this approach employ predefined tasks to ensure that users experience specific aspects of the system, whereas other studies use no task assignments. Yet the effect of using predefined tasks is still to be uncovered. There is also limited research on instructions for users in identifying usability problems. This paper reports from a comparative study of the effect of task assignments and instruction types on the problems identified in remote asynchronous usability testing of a website for information retrieval, involving 53 prospective users. The results show that users solving predefined tasks identified significantly more usability problems with a significantly higher level of agreement than those working on their own authentic tasks. Moreover, users that were instructed by means of examples of usability problems identified significantly more usability problems than those who received a conceptual definition of usability problems.

Author Keywords
Remote testing; usability testing; asynchronous testing; task assignments; instruction types; empirical study.

ACM Classification Keywords
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *Evaluation/methodology, Theory and methods.*

INTRODUCTION
Remote usability testing has been promoted as a solution to key logistic challenges in conventional usability testing, because it enables evaluators to be "separated in space and/or time from users" [7]. The first methods were presented about 15 years ago, and some empirical studies were conducted [13]. More recently, it has been demonstrated that remote synchronous usability testing performs similarly to conventional user-based testing in a laboratory [25] while reducing main logistical challenges, most notably the need to get users to the lab [4, 12]. Unfortunately, the synchronous approach does not resolve the classical challenge of traversing hours of video recordings to identify usability problems [19].

The asynchronous approach moves remote usability testing a step further by involving prospective users. They identify and describe usability problems while using the system that is being tested. The asynchronous approach has strong advantages; in particular it reduces the time spent by the evaluators considerably. Yet there are still challenges that must be resolved before it can produce results that are comparable to conventional and remote synchronous approaches [2, 5]. Here, we focus on two challenges.

First, most studies of remote asynchronous methods employ task assignments to ensure that the users experience certain aspects of the system [2, 5, 6, 7, 12]. Yet considerable knowledge of the usage domain is needed to define good task assignments [11, 16], and predefined tasks compromise validity, because users are forced into artificial usage situations [3]. Alternatively, users could work with their own authentic tasks while reporting usability problems.

Second, some type of instruction is needed to enable users to report usability problems. Training users in this complex task is challenging. Early studies of remote asynchronous testing involved extensive training with instructors and users physically present together [6, 7, 28]. Yet that solution contradicts the whole idea of remote testing. Alternatively, users could receive written instructions over the Internet [2].

This paper presents a comparative study with the aim of examining the effect of task assignments and instruction types on the number of identified problems and problem variability within a remote asynchronous usability test. In the following section, we provide the theoretical background for the study and formulate five hypotheses for the study. Then we present an overview of related work on remote asynchronous testing related to these hypotheses. This is followed by a description of the experimental method applied in the study. The next section presents the results of the study. Then we discuss the findings and compare them to related work. Finally we provide the conclusion.

CHI'12, May 5–10, 2012, Austin, Texas, USA.

BACKGROUND AND HYPOTHESES

This section discusses practical guidance and theories related to task assignments and instruction types. The discussion is summarized in a set of hypotheses.

Task Assignments

For many years one of the most disputed guidelines related to usability testing in practice is the number of users needed to achieve a satisfactory outcome [17]. Lindgaard and Chattratichart analysed usability reports written by nine professional usability teams that tested the same interface. Their findings show no significant correlation between the number of users and the number of severe problems identified. However, there was a significant correlation between the number of tasks and the number of problems identified where higher task coverage causes a higher number of problems to be identified [20]. This indicates that task coverage has more impact on the number of identified usability problems than the number of test users.

Hertzum and Jacobsen reviewed eleven research papers to examine whether the impact of the evaluator effect could be dismissed as mere chance. The results in reviewed studies indicate that a vague goal analysis, which includes design of task assignments, causes increased variability in the set of problems identified by evaluators [13]. This in turn leads to a low any-two agreement.

Based on these research results, we have defined the following hypotheses regarding task assignments:

H1: Conditions with higher task coverage reveal more usability problems than conditions with lower task coverage.

H2: Conditions with no predefined tasks introduce more variability in identified usability problems compared to conditions based on predefined tasks.

Instruction Types

When users become directly involved in usability testing, they need training in identifying usability problems. If the training requires the users to come to a physical meeting, like in [6, 7, 28], it defies the very purpose of remote usability testing. There is little research on which instructions to apply as no previous studies have focused explicitly on the effect of different types of instructions in a remote setting. It is more in line with the idea of remote testing, if the instructions are delivered over the Internet, and they must be clear without being burdensome as users are interested in getting their work done as quickly as possible [7, 15, 26]. In our study, we distinguish between deductive and inductive instructions [9].

Deductive instructions reflect a classical way of conveying information, e.g. in engineering and science [24]. The teacher presents a general rule (or definition) to be learned, after which the learners reason on observations or examples that fit within the rule. Thus learners are told up front exactly what they need to know, which makes it a straightforward and well-structured approach to teaching.

Inductive instructions cover several approaches, including problem-based learning and discovery learning, which share the same underlying principle [24]. Specific observations or examples are presented initially, and then learners infer the general rule [24]. The examples must be familiar to the learners in order to create the best possible conditions for them to assimilate the new knowledge within their existing knowledge structures [24]. It has been argued that inductive instructions motivate learners to a greater degree than traditional deductive instructions hereby making induction more effective with respect to learning outcome [9, 24]. However, inductive instructions may also cause students to infer the wrong rule or the rule may be too narrow in its application [29]. This is a consequence if the examples are too few, narrow and not concise enough [9].

Combined deductive and inductive instructions is the third option. Some learners are best stimulated by deductive instructions while others prefer induction [9], thus an obvious idea is to combine them to stimulate both types.

Based on these theoretical considerations, we have defined the following hypotheses regarding instruction types:

H3: A) Inductive instructions cause users to identify more problems than by deduction. B) Instructions based on a combination of deduction and induction cause users to identify more problems than those receiving pure inductive or deductive instructions.

H4: Inductive instructions introduce the bias that users identify problems of the same category as the examples.

H5: Instructions based on a combination of induction and deduction will be preferred over the individual types.

RELATED WORK

This section presents related empirical work on remote asynchronous methods. The discussion is structured by our focus on the use of task assignments and the instruction types employed to train users in identifying usability problems. Table 1 provides an overview of the literature.

	Task assignments	
Instruction type	**Tasks**	**No tasks**
Deductive		
Inductive	6, 7, 12	
Combined deductive and inductive	5	
Not explicit	2, 21, 28	1, 3
No training	30, 31	27

Table 1: Overview of related work categorized according to our focus on task assignments and instruction types.

Three papers, based on the same study, describe training of users with an inductive approach based on specific

examples. This was done through a video presentation and hands-on exercises where the users were physically present [6, 7, 12]. In relation to task assignment, users were given 6 tasks to solve while participating. The objective in this study was to develop a remote asynchronous method, known as user-reported critical incidents (UCI), and investigate its feasibility. The effectiveness of an example-based video and hands-on exercise is compared between two groups of users. The type of instruction did not reveal any differences with respect to the number of usability problems identified.

A single study based user training on a combination of deductive and inductive instructions by providing learners with a definition of what a usability problem is and then showing examples [5]. The experiment required the users to solve 9 tasks. Training was done through written instructions sent to the users over the Internet.

Three studies do not state explicitly which instruction type they employ for user training. All three experiments are based on users solving tasks with given systems. Two of these study the effectiveness of the UCI method compared to a conventional lab setting [2, 28]. In [2], users were given 9 tasks to solve. In [28], the total number of tasks given is not mentioned. The final paper in this category describes a comparative study of a remote asynchronous method and a laboratory based method [21]. In this case the focus is not the UCI method but rather on a more open-ended reporting format. The users solved 5 tasks.

Two studies are not based on task solving. They are not explicit on the instruction type used. One of these focuses on evaluating the feasibility of the UCI method by comparing this to laboratory testing [3]. The other paper in this category is a comparative study of a remote asynchronous method and laboratory and expert inspection methods [1]. That paper does not study the UCI method but focus on a more open-ended reporting format.

Two papers describe remote usability studies in which users did not receive training in problem identification. In [31], auto logging, used for a formative test, is compared to of a traditional lab test. Yet the number of tasks is not reported. In [30], the effect of letting users fill in closed or semi-closed questionnaires based on customized frameworks was studied. The users solved 17 tasks.

In the last study, the users did not receive any training in identifying usability problems nor did they solve any tasks while testing systems. The users filled in a closed questionnaire based on a customized framework [27].

The research reviewed above primarily comprises feasibility studies of the performance of remote asynchronous methods compared to user-based laboratory testing or expert inspections. Only one of the studies focuses on different instructions. They compare an example-based video and a set of hands-on exercises [6]. This is, however, outside our aim of comparing inductive, deductive and combined instructions, because both of their types of are inductive and users were physically present during the training sessions. There is more research on task assignments as 9 papers provide tasks for users to solve while 3 papers do not. However, none of them compare conditions with and without task assignments.

METHOD

This section describes the method of our experiment. It included six remote asynchronous conditions. A conventional laboratory-based test [25] was used as a benchmark as this is common practice in usability testing research. The seven conditions are described below and summarized in Table 2.

	Tasks	**No tasks**
Deductive	DT (n=8)	DN (n=8)
Inductive	IT (n=7)	IN (n=6)
Deductive & Inductive	DIT (n=8)	DIN (n=6)
Laboratory testing	LAB (n=10)	

Table 2: Overview of the seven conditions (n=number of participants).

Task Assignments

The use of task assignments was one independent variable in the experiment.

Tasks. The test participants in three of the remote conditions received nine predefined task assignments to solve while using the system. These conditions are denoted as DT, IT and DIT (T for 'Task') and appear in the left hand column of Table 2. The tasks were derived from an interview with a manager and a secretary from the application domain. Each participant received a list with the nine tasks appearing in randomized order. The participants were asked to solve these tasks within four weeks and report the usability problems they had experienced.

No tasks. In the other three remote conditions, no predefined tasks were given. These conditions are denoted as DN, IN and DIN (N for 'No task') and appear in the right hand column of Table 2. The participants were asked to report the usability problems they experienced during their daily use, i.e. when using the system for their own purposes, and do it within a timeframe of four weeks.

Instruction Types

The instruction type was the other independent variable in the experiment. In all remote conditions, the participants received written instructions for training them in identification of usability problems. Above, it was emphasized that instructions provided remotely must be simple [7, 15, 26]. Therefore, the instructions were limited to a half page.

Deductive instructions. The test participants in two of the remote conditions received deductive instructions. These conditions are denoted as DT and DN (D for 'Deductive')

and appear in the first row of Table 2. For these conditions, we devised a purely deductive instruction by providing the general rule in the form of a conceptual definition of what a usability problem is. The definition was an inverted definition of a usable system as defined by Molich, cf. [22], combined with 1-2 lines of clarification for each of the following elements: "Not useful", "Difficult to learn", "Difficult to remember", "Ineffective to use" and "Unsatisfying to use".

Inductive instructions. The test participants in two of the remote conditions received inductive instructions. These conditions are denoted as IT and IN (I for 'Inductive') and appear in the second row of Table 2. For these conditions, we provided examples of usability problems but no definition. The participants were expected to use the examples to derive the general rule of what a usability problem is. We gave two examples of usability problems related to consistency and affordance; cf. [23] for description of these categories of usability problems. We chose to provide one example from Facebook and one from MS Word as the participants were familiar with these systems. According to the theory of inductive learning, these examples could fit into existing mental structures.

Combined deductive and inductive instructions. The test participants in two of the remote conditions received a combination of the deductive and inductive instructions. These conditions are denoted as DIT and DIN (DI for 'Deductive and inductive') and appear in the third row of Table 2. For these conditions, we combined the deductive and inductive instructions to form one page of instructions.

System

The system that was tested in the experiment was a website for a school in a university that offers a range of educations in information technology (www.sict.aau.dk). The website provides information about study regulations, educations, exams, study board members, contact information, campus maps etc. The majority of functionality on the core website enables students to retrieve information about educations, organization etc., and that is the part we tested. There are links to other websites with more interactive functions that we did not test, e.g. signing up for exams.

Participants

The participants in the experiment were students from various graduate and under-graduate educations in computer science, software engineering, informatics and other ICT related areas. Recruitment was done via an online screening survey sent to all students in the school. We promised the students that their participation in the experiment would only take a limited amount of time (a total of 1 hour). Through the survey, we collected demographic information such as education, age, sex and experience in using the system to be tested.

73 students responded to the survey. To limit the effort for each student, we decided to conduct a between-subjects experiment. They were evenly distributed over all conditions based on their demographic profiles.

53 participants completed the experiment. Thus the dropout rate was 24%, which is the same drop-off rate as reported in [30]. Table 2 shows the distribution of the 53 participants on the seven conditions. All participants received a gift.

Procedure for the Remote Conditions

For the remote conditions, we decided to use the UCI method, because it had demonstrated the best performance among existing remote asynchronous methods [5].

The participants received instructions as described above. Reminders were sent once every week, and participants who had not reported any problems were sent a reminder before the end of the period. This was done because a previous study has described problems with participants not reporting, which was attributed to the researchers failing to send reminders during the experiment [18].

Setting

The participants were not required to work with the system in a specific setting. They worked at home or at the university using their own computers.

Procedure and Data Collection

In accordance with the UCI method, participants were instructed to report any usability problem they found on the website as soon as they discovered it. This was done using a web-based report form that was programmed using PHP, JavaScript and a MySql database. The participants received a unique login and a link to the online report form. The participants in the DT, IT and DIT conditions also received their list with task assignments.

When the participants logged in, they were presented with the instructions pertaining to their specific condition. When they had finished reading, they pressed the "Start" button and were redirected to the report form. The form was similar to that used in other UCI experiments [5, 6, 7, 12].

The following points had to be answered using this form: Task (only for the IT, DT and DIT conditions), title of the webpage in which the problem occurred, intention, expectation, problem description, problem work-around and problem severity. At the bottom of the form, there was a submit button. When it was pressed, the data were saved in the MySQL database and the form was reset, ready for a new entry. The form was running in a separate browser window, so the participants toggled between the windows each time they encountered a problem.

Procedure for the Laboratory Condition

For the lab condition, we conducted a conventional user-based testing, cf. [25]. It was not conducted by the authors of this paper.

Setting

The test was conducted in a state-of-the art usability laboratory. In the test room, the test participant sat in front of the computer and next to her/him was a test monitor whose primary task was to ensure that the test participant was thinking aloud.

Procedure and Data Collection

The test participants were introduced to the test sequence and the concept of thinking aloud by the test monitor. We scheduled one hour per participant. The participants had to solve the nine tasks that were used in the DT, IT and DIT conditions while thinking aloud. Each participant received the nine tasks in randomized order. A video of the computer's desktop and a small picture of the participants' face were recorded.

Data Analysis

The data analysis was conducted by the two authors of this paper and four external evaluators who did not otherwise take part in the experiment. All six analysed the data from the remote conditions and three of them also analysed the video material from the lab test.

All data was collected before conducting the analysis. It consisted of 53 data sets (10 videos and 43 problem reports). Each data set was given a unique identifier, and a random list was generated for each evaluator, defining the order of analysis for all data sets. This was done to reduce an ordering bias. Each evaluator analysed all the data sets alone, one at a time. For the lab condition, the videos were thoroughly analysed trough a classical video analysis.

The data sets from the six remote conditions were analysed by reading one problem report at a time. By using only the information available in the users' problem description, it was transformed into a usability problem description. If necessary, the website was checked to get a better understanding of the problem.

Problem reports from the remote conditions were validated by considering the comprehensiveness of the wording, i.e. that the problem was formulated in such a way that we could understand the problem and locate it in the user interface. This is similar to the validation procedure described in [3]. If a description could not be translated into a meaningful problem in short time or the problem could not be identified using the website, the problem was not included in the problem list. During the analysis, the evaluators also rated the severity of the problems by using the categories of critical, serious and cosmetic.

Each evaluator created a problem list containing problems from all data sets. These lists were merged to form a joint problem list for all evaluators. In case of disagreement, it was negotiated by referring to the website and the original data until agreement was reached. Severity ratings in the joined lists were made by using the most serious rating. The resulting problem list included a detailed description of each usability problem.

The evaluators identified 42 usability problems in total (15 critical, 15 serious and 12 cosmetic). Thirty-eight instances from the UCI problem reports could not be turned into problems as they were impossible to understand. Of these, 6 were from the DT condition, 9 from DN, 10 from IT, 5 from IN, 3 from DIT and 5 from DIN. A Kappa inter-rater reliability measure shows a fair agreement ($0.4 <= p <= 0.56$) between evaluators on these 38 instances [10]. This agreement may seem low, but for a complex task like usability problem identification, it is actually at the better end of the scale [13].

RESULTS

In this section, we present our findings on the effect of using task assignments and different instruction types on the outcome of a remote asynchronous usability testing.

Effect of Task Assignments

The left hand box plots in Figure 1 provides an overview of the number of tasks solved per participant in pooled non-task based and task based conditions. For the task-based conditions we asked all participants to solve the 9 tasks provided. However, in case of non-task based conditions, we have no information about the total number of tasks that each participant tried to solve. An approximation can be derived through the data given in participants' problem descriptions, but this does not include information on whether a task was attempted solved, but no report of problems was generated. To make a fair comparison between the task based and non-task based conditions, we base all numbers of solved tasks only on the data given in participants' problem descriptions.

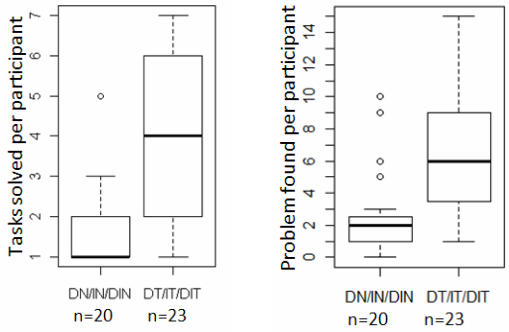

Figure 1: Left - Number of tasks attempted solved per participant (reported in problem descriptions). Right - Number of problems found per participant. The circular points indicate outliers.

The participants in the task-based conditions (DT/IT/DIT) attempted to solve a mean of 3.8 tasks (SD=1.96, n=23). In the non-task based (DN/IN/DIN) each participant attempted to solve a mean of 1.5 tasks (SD=1.04, n=19) when the outlier is removed (the circular point in Figure 1). A two-

sample t-test reveals a highly significant difference (t=4.97, df=38, p<0.001) between these conditions.

The right hand box plot in Figure 1 provides an overview of the number of problems identified. It shows that subjects in task-based conditions (DT/IT/DIT) generally identify more problems than in the non-task based (DN/IN/DIN). By removing the four outliers, each participant in DN/IN/DIN on average finds fewer problems (μ=2.16, SD=2.48, n=16) than in the DT/IT/DIT conditions (μ=6.67, SD=3.82, n=23). A two-sample t-test indicates that this difference is highly significant (t=-4.67, df=39.644, p<0.001). The number of problems found per task reveals that users in the remote task based conditions identifies a mean of 1.83 problems per task (SD=0.62) and that users in the non-task based settings on average finds 1.25 problems per task (SD=0.94). In this respect a two-sample t-test reveals no significant difference on this matter (t=-2.33, df=29.364, p>0.02).

	Critical	Serious	Cosmetic	Total
DN/IN/DIN n=20	4	6	3	**13 (31)**
DT/IT/DIT n=23	13	10	6	**29 (69)**
LAB n=10	12	13	11	**36 (86)**
Total	**15**	**15**	**12**	**42 (100)**

Table 3: Total number of problems found by pooling remote conditions with identical use of task assignments (n=number of participants). Numbers in parenthesis indicate percentage of the total.

Table 3 provides an overview of the total number of problems identified within pooled task based, non-task based conditions and LAB. By giving remote participants task assignments they are able to identify a total of 29 problems (69%) of which 13 are critical, 10 serious and 6 cosmetic. By not providing any predefined tasks users report a total of 13 problems (31%) when the four outliers shown in the right hand box plot on Figure 1 are removed. Of these problems, 4 are critical, 6 serious and 3 cosmetic. In the LAB condition 36 problems were identified (86%) of which 12 are critical, 13 serious and 11 cosmetic.

A Fishers exact test reveals highly significant differences in the total number of problems identified between remote task based and non-task based conditions (df=1, p<0.01). This also applies when comparing LAB and non-task based. However, we see no significant differences between remote task based conditions and LAB (df=1, p>0.1). This shows that by solving more tasks, participants identify significantly more problems. Yet, the number of problems found per task is similar between users in task based and non-task based conditions.

Problem Agreement

Table 4 shows the mean any-two agreement between users reporting problems in each of the remote conditions and the three evaluators who analysed video data in the LAB condition.

DT (n=21)	DN (n=36)	IT (n=28)	IN (n=6)	DIT (n=45)	DIN (n=15)	LAB (n=3)
0.24 (0.17)	0.06 (0.2)	0.33 (0.14)	0.02 (0.06)	0.27 (0.19)	0.05 (0.12)	0.54 (0.1)

Table 4: Mean any-two agreement. Parentheses indicate standard deviations (n=number of unique pairs of users/evaluators).

The highest any-two agreement is seen between the three evaluators in the conventional LAB condition (μ=0.54, SD=0.1). The remote conditions have lower agreement; user agreement in all the task based conditions is higher (μ=[0.24;0.27;0.33], SD=[0.14;0.17;0.19]) than between users in non-task based (μ=[0.02;0.05;0.06], SD=[0.06; 0.12;0.2]). A one-way ANOVA test of all means shows significant differences between one or more of the conditions (df-resid=147, F=13.09, p<0.001). A Tukey's pair-wise comparison test reveals significant difference between all task based and non-task based conditions where 0.001<p<0.03. There is no significant difference between the LAB and remote task based conditions. Thus, the non-task based conditions cause users to have significantly less overlap in identified problems than in the task based and LAB conditions.

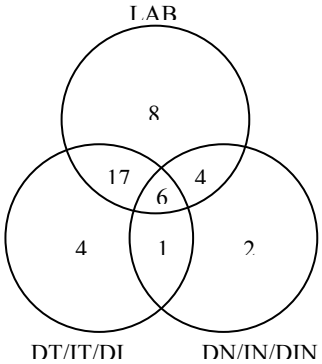

Figure 2. Venn diagram illustrating problem agreement between LAB and remote conditions pooled according to identical use of task assignments.

The agreement on the total number of usability problems identified across all conditions is illustrated in Figure 2. The pooled remote conditions identify 34 problems (the two bottom circles) with an agreement on 7 (21%). The remote task based conditions reveal 21 problems (61%) not found by the non-task based, while the latter uniquely identify 6 problems (18%). A Kappa inter-rater measure shows a poor agreement between remote task-based and non-task based conditions (p<0.4) [10].

The non-task based conditions and LAB together (the bottom right and top circle) identify a total of 38 problems. In this case 10 problems are agreed upon (26%), the non-task based conditions uniquely find 3 problems (8%) and

LAB uniquely finds 25 problems (66%). This corresponds to a poor Kappa agreement (p<0.4).

The LAB and remote task based conditions together (the bottom left and top circle) identify a total of 40 problems with an agreement on 23 (58%). The remote task based conditions reveal 5 problems not found by LAB (12%), while LAB identifies uniquely identifies 12 problems (30%). This corresponds to a good Kappa agreement (0.57<=p<=0.75). Thus, findings reveal a good agreement between task-based settings and a poor agreement between task-based and non-task based.

Effect of Instruction Types

The box plot in Figure 3 shows the number of problems identified per participant when pooling conditions with identical instruction types. Users who received deductive instructions identified fewer problems on average (μ=3.13, SD=3.67, n=16) than users receiving inductive instructions (μ=6.01, SD=3.26, n=13) or combined instructions (μ=5.31, SD=4.32, n=14).

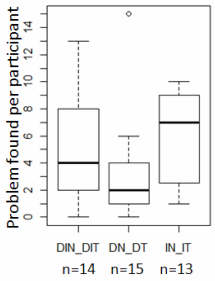

Figure 3: Number of problems found per participant in conditions pooled according to identical instruction type. The circular point indicates an outlier.

When removing the outlier in the DN/DT condition (the circular point in Figure 3), a one-way ANOVA test of all user means reveals significant differences between one or more of the pooled conditions (df-resid=40, F=4.968, p<0.02). A Tukey's pair-wise comparison test indicates significant difference between the DN/DT and DIN/DIT conditions (p<0.05) as well as DN/DT and IN/IT (p<.0.02). The difference between the DIN/DIT and IN/IT conditions is not significant (p>0.1). Thus, the inductive or combined instruction types cause users to identify significantly more problems than instructions based on deduction.

Table 5 shows the total number of problems identified in remote conditions pooled by instruction types, and the LAB condition, when removing the outlier shown in Figure 3. For the remote conditions, the inductive instructions reveal most problems (29) compared to deductive or combined instructions that give 17 and 25 problems respectively.

A Fishers exact test reveals a significant difference in the total number of identified problems between the IN/IT and DN/DT conditions (df=1, p<0.02). There is no significant difference between IN/IT and DIN/DIT (df=1, p>0.1). Furthermore, we see a significant difference between LAB-DN/DT and LAB-DIN/DIT conditions (df=1, p<0.01). Thus conditions in which users received inductive instructions identified significantly more problems in total than conditions where users were given deductive instructions.

	Critical	Serious	Cosmetic	Total
DIN/DIT (n=14)	10	9	6	**25 (59)**
DN/DT (n=15)	9	8	0	**17 (40)**
IN/IT (n=13)	13	10	6	**29 (69)**
LAB (n=10)	12	13	11	**36 (86)**

Table 5: Total number of problems found by pooling remote conditions with identical instruction types (n=number of users). Numbers in parenthesis indicate percentage of the total.

Problem Types

We examined the types of problems, cf. [23], identified in all conditions to uncover whether users who received inductive instructions were biased towards identifying problems of the same type as in the examples provided in the inductive instruction (affordance and consistency). This revealed no significant differences.

Ratings of Instruction type

Users in the remote conditions gave satisfaction ratings of the instructions on a 5 point Likert scale (1=lowest satisfaction, 5=highest satisfaction). Table 6 shows the median ratings pooled by instruction types.

DN/DT (n=16)	IN/IT (n=13)	DIN/DIT (n=14)
4	5	4

Table 6: Median satisfaction ratings given on instruction types in remote conditions (n=number of participants).

A non-parametric Kruskal Walis test shows a significant difference between ratings for the DN/DT and IN/IT conditions (χ^2 = 4.4, df-con= 1, p<0.04) where induction is rated higher (median=5) than deduction (median=4). There is no significant difference between the DN/DT and DIN/DIT conditions (χ^2=0.32, df-con = 1, p>0.5) and IN/IT and DIN/DIT (χ^2=1.99, df-con=1, p>0.1). Thus, participants rate instructions based on induction higher than deduction or the combination of the two.

DICUSSION

In this section we discuss our findings in relation to the five hypotheses and related work.

H1: Tasks Solved and Problems Found

H1: Conditions with higher task coverage reveal more usability problems than conditions with lower task coverage.

We found that task assignments improve the outcome of remote asynchronous usability testing. Thus we accept H1. Participants in the non-task based conditions on average solved significantly fewer tasks and in turn identified significantly fewer usability problems than participants in the task based conditions. This has also been found in a conventional lab setting [20]. It should, however, be noted that the number of problems found per task is similar for the task based and non-task based conditions.

We have not found studies of remote asynchronous testing that compare users solving predefined tasks with users working on their own problems. However, there are studies of either of these options. Three studies used no predefined tasks when comparing remote asynchronous methods to conventional lab or inspection methods. Two of these report that users applying remote methods identified between 67% and 73% of all problems [1, 27]- In the third, the users found more problems than the lab condition [3]. In our study we found that users in the pooled remote non-task based conditions found 31% of all problems, which is lower than the three studies. Unfortunately, none of these studies specify how many tasks the users solved. We found that participants solving more tasks identify more problems. If the users in the three studies solved more tasks than our users, that could explain the difference.

Nine other studies of remote asynchronous testing have used predefined tasks. In four of these, users applying remote asynchronous testing found between 52% - 68% of all problems [6, 7, 12, 30]. This is comparable to our findings, where participants in the pooled task based conditions identified 69% of all problems. Five other studies report lower numbers as their users in remote task based conditions uncovered between 21%-47% of all problems [2, 5, 21, 28, 31]. Four of these latter studies are either not explicit on the instruction type applied for training users or have not provided any. This may explain the difference.

H2: Task Assignments and Problem Variability

> **H2:** Conditions with no predefined tasks introduce more variability in identified usability problems compared to conditions based on predefined tasks.

We found that lack of predefined task assignments increased the variation among the usability problems identified. Our results indicate that participants in remote non-task based conditions have a significantly lower any-two agreement compared to participants in task based conditions and evaluators in the LAB condition. Thus we accept H2. This is consistent with a study concluding that a vague goal analysis causes an increased variability in the number of usability problems identified [13].

We found a good agreement between task-based settings and a poor agreement between task-based and non-task based. Thus users who did not receive predefined tasks had significantly less overlap in identified problems. All the participants who did not receive predefined tasks attempted to solve a total of 14 tasks. Four of these were similar to four of the 9 predefined tasks given to participants in task based conditions. Thus across all users, 10 tasks were uniquely solved in non-task based conditions, which gave a focus on other areas of the system, i.e. users in these conditions saw different parts of the website. This demonstrates that authentic system use fit well with exploratory tests where specific goals are missing. On the other hand, if there are specific areas of interest in an interface, users should be given predefined tasks to keep them within these limits.

We have found five studies that report problem agreement with remote asynchronous testing. Two of these report an agreement of 20% and 31% between remote task based and lab conditions [21, 30]. A third study found an agreement of 51% between a remote non-task based condition and expert inspection [27]. These results do not correspond to ours, as we have a higher agreement between remote task based and lab conditions (58%) and lower for of non-task based (8%). These differences may be caused by variations in instruction types [21] and lack of training [30, 27]. There is a higher correlation with related work for unique problems. Three studies report that, by merging problems found via task based remote conditions and lab, the remote conditions uniquely identify between 2% - 26% of all problems [2, 5, 30]. We find similar results as, compared to the lab condition, the remote task based conditions uniquely identified 21% of the problems.

H3 – Effectiveness of Inductive Instructions

> **H3:** A) Inductive instructions cause users to identify more problems than by deduction. B) Instructions based on a combination of deduction and induction cause users to identify more problems than those receiving pure inductive or deductive instructions.

By pooling the remote conditions with identical instruction types we found that each participant on average identified significantly more problems when given inductive instructions compared to those who were given deductive. Thus we accept H3(A).

The combination of deductive and inductive instructions also caused each participant to uncover significantly more problems than those who received deductive instructions only, which support H3(B). However, there was a tendency that the users receiving inductive instructions identified more problems than those who received the combination, but this difference was not significant. The latter finding ultimately causes us to reject H3(B).

We have found four papers that explicitly describe the type of instruction given to users. Three of these are based on the same study and compare the effectiveness of two types of inductive instructions conveyed in physical presence [6, 7, 12]. The first of these is an example-based video. The second is a hands-on exercise. These two instructions are

compared between two groups of users. The results did not reveal any differences in the number of problems identified. The users in the remote conditions identified 68% of the usability problems found in a conventional video-based analysis. This is similar to our findings where participants receiving inductive instructions, identified 69% of all problems. Another study used a combination of deductive and inductive instructions by presenting a definition of a usability problem and some examples [5]. The users trained this way were able to find between 21% and 47% of all problems, depending on the remote method applied. Our result for the similar condition was 59% of all problems.

H4 –Bias towards Instruction type

> **H4:** Inductive instructions introduce the bias that users identify problems of the same category as the examples.

We did not find that inductive instructions introduced a bias by causing users to infer a wrong rule or a rule too narrow in its application. Our inductive instruction provided two examples; one of an affordance problem and one of a consistency problem. Yet our results showed no significant differences that reflected a bias towards those two types of problems. Thus we reject H4.

We have found a single paper that describes the types of problems identified by users in a remote asynchronous condition [1]. However, we do not know the instruction type applied and, therefore, we do not know whether the problems identified were the same type as eventual examples.

H5 – Subjective Preferences of Instruction type

> **H5:** Instructions based on a combination of induction and deduction will be preferred over the individual types.

We did not find that the combination of the deductive and inductive instruction types was preferred by the users. We measured the subjective satisfaction with the instruction types across the users in the remote conditions. We did not find significant differences between instructions based on a combination and instructions exclusively based on one of the types. Thus we reject H5.

This result contradict advice in the literature which states that Instructions should be based on a combination of deduction and induction as this stimulates learners preferring either type [9]. An explanation why participants did not rate the combined instructions highest may be that such instructions results in more text which causes training to be more burdensome and time-consuming, which should be avoided in remote usability testing [7, 15, 26]. None of the studies in related work present findings on user ratings of instructions, hereby making a comparison to these impossible. It is also interesting that the users who received inductive instructions identified more problems than those receiving deductive or a combination thereof.

CONCLUSION

An increasing body of research demonstrates that remote asynchronous usability testing has promising benefits. However, there are still aspects that need to be developed. In this paper, we have presented a comparative empirical study of the effect of task assignments and instruction types on the result of a remote asynchronous usability test. The study joins a trend where mere method comparisons are replaced with practice-oriented studies of the effects of variations in method use.

Our findings show that users receiving predefined tasks solved significantly more tasks, identified significantly more usability problems and had a significantly higher level of problem agreement than those working on their own authentic tasks. Not providing predefined tasks caused the users to identify more varied sets of problems than when predefined tasks were provided. Finally, users who were instructed by means of inductive examples of usability problems identified significantly more usability problems than users who were given a deductive conceptual definition, and the satisfaction rating for the inductive instruction was significantly higher.

The results are limited by the type of website tested, which is mainly used for medium frequency information retrieval. Also, the sheer size of the website implies that different users may have experienced various parts that appear differently, this is particularly likely with the non-task users. Moreover, other types of users and systems may reveal different results, for example systems applied more frequently by expert users to solve more complex tasks.

In the future it would be relevant to conduct similar experiments based on other types of systems and users. It would also be interesting to resolve the basic challenge of collecting more information about user activity in a remote asynchronous test, e.g. about the tasks they have attempted to solve and the way they have done it.

ACKNOWLEDGMENTS

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

REFERENCES

1. Äijö, R. and Mantere, J. Are Non-Expert Usability Evaluations Valuable? http://www.hft.org/HFT01/paper01/acceptance/2_01.pdf
2. Andreasen, M. S., Nielsen, H. V., Schrøder, S. O. and Stage, J. What happened to remote usability testing? An empirical study of three methods. In *proc. CHI 2007*, ACM Press (2007), 1405-1414.
3. Bosenick, T., Kehr, S., Kühn, M. and Nufer, S. Remote usability tests: an extension of the usability toolbox for online-shops. In *Proc. UAHCI 2007*, Springer-Verlag (2007), 392-398.

4. Brush, A. B., Ames, M. and Davis, J. A comparison of synchronous remote and local usability studies for an expert interface. In *proc. CHI 2004*, ACM Press (2004), 1179-1182.
5. Bruun, A., Gull, P., Hofmeister, L. and Stage, J. Let Your Users Do the Testing: A Comparison of Three Asynchronous Usability Testing Methods. In *proc. CHI 2009*, ACM Press (2009), 1619-1628.
6. Castillo, J. C. *The User-Reported Critical Incident Method for Remote Usability Evaluation*. Master thesis, Virginia Polytechnic Institute and State University (1997).
7. Castillo, J. C., Hartson, H. R. and Hix, D. Remote usability evaluation: Can users report their own critical incidents? In *proc. CHI 1998*, ACM Press (1998), 253-254.
8. Felder, R.M. Reaching the second tier: Learning and teaching styles in college science education. *College Science Teaching 23*, 5 (1993), 286-290.
9. Felder, R.M. and Silverman, L.K. Learning and Teaching Styles in Engineering Education. *Engineering. Education 78*, 7 (1988), 674-681.
10. Fleiss, J.L. *Statistical methods for rates and proportions (2nd ed.)*. John Wiley & Sons, New York, 1981.
11. Følstad, A. and Hornbæk, K. Work-domain knowledge in usability evaluation: Experiences with Cooperative Usability Testing. *Journal of Systems and Software 83*, 11, (2010), 2019-2030.
12. Hartson, H. R. and Castillo, J. C. Remote evaluation for post-deployment usability improvement. In *proc. AVI 1998*, 22-29.
13. 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.
14. Hertzum, M. and Jacobsen, N.E. The Evaluator Effect: A Chilling Fact about Usability Evaluation Methods. *Human Computer Interaction 15*, 1 (2003), 1336-1340.
15. Hilbert, D.M. and Redmiles, D.F. Separating the Wheat from the Chaff in Internet-Mediated User Feedback Expectation-Driven Event Monitoring. *ACM SIGGROUP Bulletin 20*, 1, (1999), 35-40.
16. Hornbæk, K. and Frøkjær, E. Making Use of Business Goals in Usability Evaluation: An Experiment with Novice Evaluators. In *proc. CHI 2008*, ACM Press (2008), 903-911.
17. Hwang, W. and Salvendy, G. Number of people required for usability evaluation: the 10±2 rule. *Commun. ACM* 53, 5 (May 2010), 130-133.
18. Kjaer, A., Madsen, K.H. and Petersen, M.G.. Methodological Challenges in the Study of Technology Use at Home. In *proc. HOIT 2000*, Kluwer Academic Publishers (2000), 45–60.
19. Kjeldskov, J., Skov, M. B. & Stage, J. Instant Data Analysis: Evaluating Usability in a Day. In *proc. NordiCHI 2004*, ACM Press (2004), 233-240.
20. Lindgaard, G. and Chattratichart, J. Usability Testing: What Have We Overlooked? In *proc. CHI 2007*, ACM Press (2007), 1415-1424.
21. Marsh, S. L., Dykes, J. and Attilakou, F. Evaluating a geovisualization prototype with two approaches: remote instructional vs. face-to-face exploratory. In *proc. Information Visualization 2006*, IEEE (2006), 310-315.
22. Molich, R. *Usable Web Design*. Nyt Teknisk Forlag, Odense, Denmark, 2007.
23. Nielsen, C. M., Overgaard, M., Pedersen, M.B., Stage, J. and Stenild, S. It's Worth the Hassle! The Added Value of Evaluating the Usability of Mobile Systems in the Field. In *proc. NordiCHI 2006*, ACM Press (2006), 272-280.
24. Prince, M.J. and Felder, R.M. Inductive teaching and learning methods: Definitions, comparisons, and research bases. *Engineering Education 95* (2006), 123-138.
25. Rubin, J. and Chisnell, D. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. Wiley Publishing, Indianapolis, USA, 2008.
26. Scholtz, J. A case study: developing a remote, rapid and automated usability testing methodology for on-line books. In *proc. HICSS 1999*, IEEE (1999).
27. Ssemugabi, S. and Villiers, R.D. A comparative study of two usability evaluation methods using a web-based e-learning application. In *proc. SAICSIT 2007*, ACM Press (2007), 132-142.
28. Thompson, J. A. *Investigating the Effectiveness of Applying the Critical Incident Technique to Remote Usability Evaluation*. Master thesis, Virginia Polytechnic Institute and State University, 1999.
29. Thornbury, S. *How to teach grammar*. Pearson Education Ltd, Harlow, Essex, England, 1999.
30. Tullis, T., Fleischman, S., McNulty,M., Cianchette, C. and Bergel, M. An empirical comparison of lab and remote usability testing of web sites. http://home.comcast.net/~tomtullis/publications/RemoteVsLab.pdf
31. Waterson, S., Landay, J. A. and Matthews, T. In the lab and out in the wild: remote web usability testing for mobile devices. In *proc. CHI 2002*, ACM Press (2002), 796-797.
32. Winckler, M. A. A., Freitas, C. M. D. S. and de Lima, J.V.Remote usability testing: a case study. In *proc. OzCHI 1999*, CHISIG (1999).