# Remote Synchronous Usability Testing as a Strategy to Integrate Usability Evaluations in the Software Development Process: A Field Study

Fulvio Lizano
Informatics School
National University
Heredia, Costa Rica
flizano@una.cr

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

***Abstract*—Although Human-Computer Interaction techniques, as usability evaluations, are considered strategic in software development, there are diverse economic and practical constraints in their application. The integration of these tests into software projects must consider practical and cost-effective methods such as, the remote synchronous testing method. This paper presents results from a field study designed to compare this method with the classic laboratory-based think-aloud method in a realistic software development context. Our interest in this study was to explore the performance of the remote synchronous testing method in a realistic context and its effectiveness to provide an integration method of usability evaluations into the software development process. The results demonstrate that the remote synchronous testing method allows the identification of a similar number of usability problems achieved by conventional methods at a usability lab. Additionally, the time spent using remote synchronous testing is significantly less. Results obtained in this study also allowed us to infer that when using the remote synchronous testing method, it is possible to handle some practical constraints that limit the integration of usability evaluations into software development projects. In this sense, the relevance of the paper is based on the positive impact that remote synchronous testing could have in the digital accessibility of the software, by allowing the extensive use of usability evaluation practices in software development projects.**

***Keywords - usability evaluations, remote synchronous testing method, integration of usability evaluation in software development projects, field study.***

## I. Introduction

In this extended paper, we improve the work presented at the eighth International Conference on Digital Society (ICDS 2014) in Barcelona, Spain, in March 2014 [1].

In this extended paper, we improve the introduction section with more elements that reflect on the relevance of usability evaluations. We have also better enriched and structured, the related work section. Special mention is deserved regarding additional references related to economic limitations (i.e., the cost obstacle) and the conceptual background on the remote synchronous test. In the method section, we included two new sub-sections. First, there is a new subsection called “The context of the study”, which presents the context where the study was made. Second, we included a subsection called “Overcoming the limitations of the field study”, where we present the main theoretical concepts of these kinds of studies. In addition, we also explain how we intended to handle such limitations. In this section we also included several figures. The result and discussion sections were also structured more clearly, with some additions. We have also enhanced the conclusion section and separated the limitations of the study into a new section. Our intention is to bring a final reflection on the main limitation presented in our study, which is related to the type of participants. Finally, the bibliographic references were extended.

Usability has a significant impact on software development projects [2]. Common usability activities, such as usability evaluations, are relevant and strategic in diverse contexts, such as users, software developers, development organisations, and software development projects.

In the case of the user, who requires a high level of usability in the software [3], usability evaluations are important because they assess whether the software under evaluation considers users’ skills, experiences, and expectations [4]. A high level of usability in a software system enables users to perform their work while saving time and resources, and this allows them to be more effective and efficient.

In the case of software developers, usability evaluations provide them with clear details about usability problems in a software system. This information becomes valuable feedback [5], which allows them to produce better results in their work. Furthermore, improved usability in software increases the developers' confidence levels regarding their technical ability and creates a personal identification with a software product; these are strong motivators for developers [6][7].

For development organisations, usability evaluations are important because they provide benefits, such as cost savings, increased sales, increased productivity, lower training costs, and decreased technical support requirements for users [8]. More usable software implies less user support and training, which increases the development organisational efficiency and productivity.

Finally, in software development projects, Human-computer Interaction (HCI) techniques have a high valuation [2]. In fact, Abran et al. [9] considered usability evaluations to be relevant and strategic activities within software projects. One of the main reasons for this high valuation is due to the application of usability methods (e.g., usability inspection methods, usability testing with users, etc.), and it is possible to improve the quality of the software by providing useful feedback about usability.

The importance of usability in the above-cited cases has motivated the integration efforts of usability evaluations into software projects [10][11][12].

However, economic and practical issues limit integration of usability evaluations into software projects, where limited schedules and high expectations of stakeholders to obtain effective/efficient results faster, are common. Productivity has been a recurrent concern in the industry [13][14] and is something that makes it very difficult to justify certain HCI activities [15].

Considering this, any effort to integrate usability evaluations into software projects must necessarily consider practical and cost-effective methods. Many of the studies conducted to explore this integration have been made on limited realistic contexts (e.g., literature reviews [15][16][17][18][19], surveys [2][13][20][21][22], experiments in labs [23][24] and case studies [5][25]). Other papers cited above present proposals of projects or methods [12][26][27]. There are only three studies with a more empirical base in more realistic contexts [5][28][29]. Confidence in the results of these studies should be improved by other studies made in a realistic developmental context.

This is the reason we present the results of a field study that aimed to compare the remote synchronous test method against the classic laboratory-based think-aloud method in a realistic software development context.

In the following section, we offer an overview of related works. The next section presents the method used in our research. Following this, we present the results of our study. After the results are summarised, the paper presents the analysis before concluding with suggestions for future work.

## II. Related works

This section presents the economic and practical constrains to integrate usability evaluations into software development projects. The literature on practical constrains included here considers only studies focused on the perspectives between practitioners, methods, and user participation. On the other hand, in this section we present the main concepts of remote synchronous usability testing. Our purpose in the literature review is to provide a basic framework to analyse the results of our study.

### A. *Economic constrains: the cost obstacle*

High consumption of resources in usability evaluations, also known as the cost obstacle, is a recurrent perception in diverse contexts [17][20][22][23][30]. This fact could explain why usability has a lower valuation for an organisation's top management [12], becoming manifest by the lack of respect and support for usability and HCI practitioners [21]. Therefore, cost-justification of usability may be difficult for many companies, as it is perceived as an extra cost or feature [15].

It is possible to define the cost obstacle as the constraints of applying usability evaluations due to the high consumption of resources required by this kind of testing. The cost of usability evaluations is a measurable obstacle presented in both of the following cases: development organisations and software projects.

In the case of development organisations, this obstacle is presented in the form of a 'perceived cost obstacle'. In this case, the perception can be understood as the perspective that development organisations have regarding the cost of the usability evaluations. This perspective is normally based on the value judgment presented within development organisations. Some examples of this modality of the cost obstacle were reported in [20] and [30], where it is possible to see that in development organisations there exists the idea that usability testing is expensive, and this limits its application– even though such evaluations have not been conducted. In addition, Nielsen [23] argues that the perception of the cost of usability engineering techniques is the reason why such techniques are not used extensively in development organisations. Coincidentally, Bellotti [31] reports that software developers view usability methods as too time-consuming and intimidating in their complexity.

Within software projects, the cost obstacle appears in the form of an 'actual cost obstacle'. Considering the dynamic presented in the software development project, based on a specific product (i.e., the software), the cost obstacle is more tangible and is related to the real cost presented in such a project. Nielsen [23] offers some examples of actual costs. Ehrlich and Rohn [32] referred to the actual costs in terms of 'initial costs' and 'sustaining costs'. The initial costs include the settings and laboratory or similar facility equipment that are required for usability evaluations. The sustaining costs correspond to those costs related to the conduction of the usability evaluation process and include the staff, recruitment of participants, transportation, allowances, special equipment, software, and etc.

The cost obstacle can be quantified by defining and collecting information about diverse usability metrics. Time consumption in usability evaluations is one of the most commonly used measures to assess cost [23][33][34][35]. Time consumption relates some ideas about the consumption of resources in usability evaluations. For example, based on this measure, some studies concluded that classical protocols, such as 'thinking aloud', have a high consumption of time [36][37]. In addition, Kjeldskov and Graham [38] found that the analysis of the data collected during the usability evaluations normally demands a high time consumption, especially in the video data analysis process. Finally, Borgholm and Madsen [39] argue that usability reports could be impractical due the extensive time used in their preparation.

It is possible to identify the following two main strategies for reducing the cost of usability evaluations: 1) use of alternative usability evaluation methods, and 2) the improvement of the usability evaluation process.

Alternative usability evaluation methods aim to reduce costs in classical usability evaluations with users. One example of these methods is the heuristic evaluation, which is a method where the software interfaces are evaluated based on usability heuristics in order to generate an opinion about the usability of the software [40]. The process starts with individual reviews by three, four, or five expert evaluators of the software. During this process, each evaluator checks whether the usability principles, used as a reference for good practices (i.e., heuristics), are included in the software. Next, evaluators compare their results and produce an integral usability report [37].

The approach of improving the usability evaluation process has been widely discussed; for example, the time consumption issue within analysis activities was addressed by Kjeldskov and Graham [38]. They proposed an analysis technique called Instant Data Analysis (IDA) that is used in the analysis process of the results of the sessions with users. The aim of IDA is to conduct usability evaluations in one day, obtaining similar results to traditional video data analysis methods. Alternatively, Borgholm and Madsen [39] suggest focusing on the report of the results of the usability evaluations. These researchers found that some HCI practitioners prepared two kinds of reports with different formats and contents. The first report was oriented to the developers and provided an executive summary with information useful for their work. The second report, which is more extensive, was delivered several days after the evaluation for documentation purposes. Supplementary meetings, at which the developers and HCI practitioners discussed the usability findings, and posters, which described the main usability problems, were used to mitigate this problem.

### B. *Other practical constrains*

On the other hand, regarding practical constrains, three of the most cited are related to the difference of perspectives between HCI and Software Engineering (SE) practitioners, the absence or diversity of methods, and user participation.

The first constraint related to the difference of perspectives between HCI and SE practitioners is contextualised in the difference of opinions regarding what is important in software development [27]. This diversity of perspectives results in contradictory points of view regarding how usability testing should be conducted and may result in a certain lack of collaboration between HCI and SE practitioners. It is possible to find the origin of this discrepancy between these two perspectives in the foundations of the HCI and SE fields. Usability is focused on how the user will work with the software, whereas the development of that software is centred on how the software should be developed in a practical and economical way [19]. These conflicting perspectives result in tensions between software developers and HCI practitioners [19][25].

The second constraint relates to the absence or diversity of methods and has two opposing views. First, some researchers report a lack of appropriate methods for usability evaluation [20][22] or a lack of formal application of HCI and SE methods [2]. This situation may explain why the User-Centred Design UCD community has expressed criticism about the real application of some software development principles [18]. Second, it is reported that the existence of numerous and varied techniques and methodologies in the HCI and SE fields could hamper the integration [25].

Finally, the participation of customers and users has become another relevant limitation for the integration of usability evaluations into software projects [20][30][22]. This matter is a permanent challenge to the dynamic of the software development process. Users and customers have their own problems and time limitations, and these normally limit their participation in software development activities, such as usability evaluations.

The literature reported different proposals for handling the aforementioned three practical constraints. First, in the case of the difference of perspectives between HCI and SE practitioners, some studies have suggested that increased participation by developers in usability testing could positively impact the valuation of usability [5]. This improvement in developer perspectives could make them more conscious of the relevance of HCI techniques.

Second, with respect to the absence or diversity of methods, an integration approach based on international standards is proposed [16] in order to enable consistency, repeatability of process, independence of organisations, quality, etc. A similar approach suggests the integration of HCI activities into software projects using SE terminology for HCI activities [26].

Finally, regarding the constraint related to the participation of customers and users, some researchers have suggested several practical actions (e.g., smaller tests in iterative software development processes, testing only some parts of the software, and using smaller groups of 1–2 users in each usability evaluation) [29].

### C. *Remote Synchronous Usability Testing*

The aforementioned obstacles can be handled by using remote synchronous usability testing, which is a method that allows software developers to conduct/participate in usability evaluations with users in a practical and economical way.

Remote Usability Testing (RUT) was defined as a usability evaluation technique in which the evaluator remains separated in space and/or time from the users while performing observation and analysis of the process [41]. The RUT techniques can be synchronous or asynchronous. The synchronous format allows the evaluators to receive and conduct the evaluation in real time with users who are located elsewhere. In contrast, in the asynchronous format, the evaluators do not access the data nor conduct the evaluation in real time [42]. The RUT method allows usability testing without the constraint of geographical limitations, and therefore requires fewer resources.

The main uses of RUT are:

- to evaluate the usability of web applications [43],
- to reduce the costs of the usability evaluation process [43][44],
- to collect a high volume of data [42], and
- to make usability evaluations by considering an international context [42].

The practicality of logistic considerations and the resource-saving advantage make RUT a promising alternative for reducing the aforementioned limitations.

## III. Method

We have conducted an empirical study aimed at comparing the remote synchronous testing method (condition

R) with the classic laboratory-based think-aloud method (condition L).

Using remote synchronous testing, the test is conducted in real time, but the evaluators are separated spatially from the users [33]. The interaction between the evaluators and users is similar to those at a usability lab. There are many studies that confirm the feasibility of RUT methods [33][45][46]. Actually, there is a clear consensus regarding the benefits obtained using this method (e.g., no geographical constraints, cost efficiency, access to a more diverse pool of users, and similar results as a conventional usability test in a lab) [33][47]. The main disadvantages are related to problems generating enough trust between the test monitor and users, longer setup time, and difficulties in re-establishing the test environment if there is a problem with the hardware or software [33].

Three usability evaluations were made by three teams using a classic usability lab. In addition, another three usability evaluations were conducted by another three teams using a remote synchronous testing method.

Final-year students of SE who had 18 months of practical experience working in software development formed all of these teams. This experience is the result of an academic project created by the students to develop a software system in a real organisation.

In the next subsections, we present in detail the main elements of the context of the study, participants, training, and advice received for the participants, procedure followed in the study, settings, data collection and analysis, and the actions taken in order to overcome the limitations of the field study.

### A. The context of the study

The study considered usability evaluations conducted on software systems made in the context of the internship/academic project for which the students put into practice what they learned in the courses of the System Engineering Bachelor degree. The organisation in charge of these academic projects is the Department of System Engineering (DSI), School of Informatics, National University (UNA), located in Heredia, Costa Rica. The UNA is one of the five public universities of Costa Rica. Funded in 1973, this university has five faculties that have an enrolment of around 18000 students in 65 undergraduate and postgraduate programs. Informatics School is the second school of the university with an approximate enrolment of 1300 students, 250 new students every year.

Starting in the third year, over three semesters, around 30 software projects are developed by student teams formed by three to four students who were also learning regular topics related to system engineering theory. In this process, the student teams receive supervisor feedback (a DSI professor) and also interact with stakeholders/users. The students design, develop, and implement a software system in a real organisation (private, public, Small and Medium Enterprise (SME), or Non-Profit Organisations (NPO)). This organisation provides regular assessments of the students' work. In addition, at the end of the process, the organisation should formally accept or reject the software product. There is a main user assigned by the organisation. This user normally plays a role as main contact between the students and the organisation. Usually, this user is also one of the main stakeholders. It is also possible that other regular users can be considered in the process. Fig. 1 presents the context around these projects.

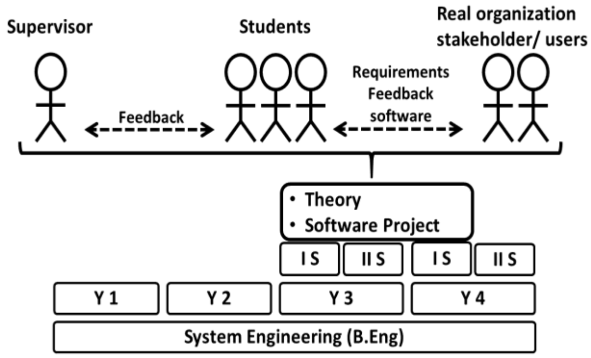


Figure 1. The context of the study: the academic project.

### B. Participants

In order to be considered for our research, the software projects must meet our requirements regarding users being available for the tests. Considering these criteria, 16 of 30 teams and their software projects were preselected as potential participants in the experiment. Finally, we randomly selected six teams who were randomly distributed throughout the R and L conditions.

Final-year students who were finishing their last course in System Engineering formed the teams. These participants were organised into six teams consisting of three members each. A total of 18 people participated in our study. The average age was 22 (SD=2.13), and 17% were female. In addition to the courses taken previously, the participants had amassed nearly 18 months of real experience in practical academic activity by developing a software system in a real organisation that sponsored the project. These organisations provided regular assessments and formal acceptance (or rejection) of the software. Several users and stakeholders were also involved in the process. The scope of the software projects was carefully controlled in order to guarantee a similar level of effort from all of the participants. The average of the final assessment of the project was 9.67 on a scale of one to ten (SD=0.33). As an incentive for participation, the participants received extra credits. The conditions, code, members, and software are presented in Table I.

### C. Training and advice

All participants received training and advice during the experiments (remotely for R condition). In the training, we presented and explained several forms and guidelines based on commonly used theories [38][47]. In addition, a workshop was created in order to put into practice the contents of the

TABLE I. TEAMS, MEMBERS, AND STAFF FOR THE TESTS

| Cond. | Code | Members | Software |
|---|---|---|---|
| L | L1 | 3 males | Students' records in a college. |
| | L2 | 1 female, 2 males | Internal postal management system in a financial department of a public university. |
| | L3 | 1 female, 2 males | Laboratory equipment management in a biological research centre belonging to a public university. |
| R | R1 | 1 female, 2 males | Criminal record in a small municipal police station. |
| | R2 | 3 males | Management of documents related to general procurement contracts in an official national emergency office. |
| | R3 | 3 males | Students' records in a public school. |

training materials. The participants received specific instructions in order to consider the following three categories of usability problems: critical, serious, and cosmetic [33]. The number of hours spent in training was ten (four hours in lectures and six hours in practice). Furthermore, the advice provided to the participants included practical issues concerning how to plan and conduct usability evaluations.

## D. Procedure

The design of the experiment increased the confidence in the results and the objectivity of the development teams during the evaluation process. Under the two conditions, each team had to test the software system made by another team, who also tested another software system made by a third team.

Each test had two main parts. The first part, under the responsibility of the team who made the software, corresponded to the planning of the complete process (e.g., planning, checklists, forms, coordination with users, general logistics, etc.). The planning included a session script with ten potential tasks of the software.

In the second part of the tests, another team conducted the sessions with the users. The test monitor of this team had to select, for each user, five tasks from those previously defined. We hypothesised that this measure would increase the impartiality of the process; the developers of the software could not interfere in the selection of the task, and the users had to work with different tasks in each session. Next, the test monitor guided the users in the development of the task while the logger and the observers took notes. The test ended with a final analysis session conducted by a facilitator [38]. Table II shows the assignment of the planning and conducting responsibilities in each test. In Fig. 2 and Fig. 3 it is possible to see examples of sessions conducted in condition L and R, respectively.

TABLE II. PLANNING AND CONDUCTION OF TESTS.

| Software made by | Planning made by | Conduction made by |
|---|---|---|
| L1 | L1 | L3 |
| L2 | L2 | L1 |
| L3 | L3 | L2 |
| R1 | R1 | R3 |
| R2 | R2 | R1 |
| R3 | R3 | R2 |

## E. Settings

The test conducted under the L condition used a state-of-the-art usability lab and think-aloud protocol. Using this technique it is possible to collect information regarding what the test participants are thinking while they perform the usability tasks. Participants are guided along the test in order to express commentaries of their thoughts by thinking aloud [8][47]. Each test included three sessions where the users were in front of the computer and the test monitor was next to the users. The logger and observers were present in the same room. In the case of the R condition, the tests were based on remote synchronous testing [33]. All participants were spatially separated. Users were in the sponsors' facilities. Each test included three sessions with the users.

## F. Data collection and analysis

Each user session was video recorded. The video included the software session recorder (video capture of the screen) and a small video image of the user. Under R conditions, the video also recorded the image of the test monitor (see Fig. 2). We also used a test log to register the main data of each activity (i.e., date, participant, role, activity, and time consumed) and the usability problem reports.

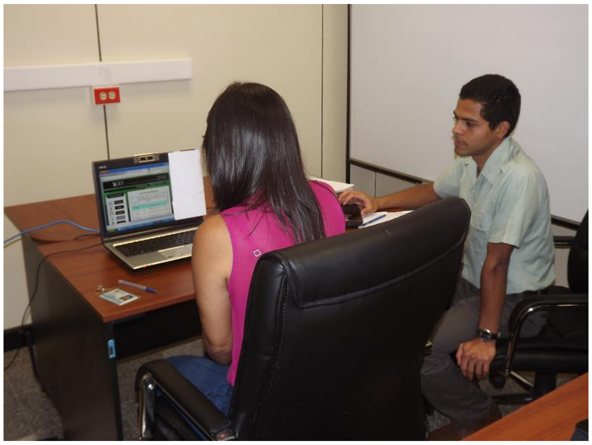
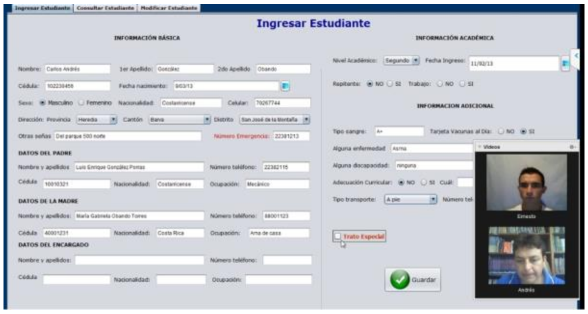


Figure 2. Example of a test's session in L condition.



Ingresar Estudiante

INFORMACIÓN BÁSICA

Nombre: Carlos Andrés 1er Apellido: González 2do Apellido: Obando
Cédula: 102230458 Fecha nacimiento: 9/03/13
Sexo: ● Masculino ○ Femenino Nacionalidad: Costarricense Celular: 70267744
Dirección: Provincia Heredia Cantón Barva Distrito San José de la Montaña
Otras señas Del parque 500 norte Número Emergencia: 22381213

DATOS DEL PADRE

Nombre y apellidos: Luis Enrique González Porras Número teléfono: 22382115
Cédula 10010321 Nacionalidad: Costarricense Ocupación: Mecánico

DATOS DE LA MADRE

Nombre y apellidos: María Gabriela Obando Torres Número teléfono: 88001123
Cédula 40001231 Nacionalidad: Costa Rica Ocupación: Ama de casa

DATOS DEL ENCARGADO

Nombre y apellidos: Número teléfono:
Cédula Nacionalidad: Ocupación:

INFORMACIÓN ACADÉMICA

Nivel Académico: Segundo Fecha Ingreso: 11/02/13
Repitente: ● NO ○ SI Trabajo: ○ NO ○ SI

INFORMACION ADICIONAL

Tipo sangre: A+ Tarjeta Vacunas al Día: ○ NO ● SI
Alguna enfermedad Asma
Alguna discapacidad: ninguna
Adecuación Curricular: ● NO ○ SI Cuál:
Tipo transporte: A pie Número tel

□ Trato Especial

Figure 3. Example of a test's session conducted in R condition.

The data analysis was conducted by the authors of this paper based on all data collected during the tests. The tests produced six sets of data for analysis (i.e., six usability problem reports, six test logs, and six videos).

The consistency of the classification of the usability problems by participants was one of the main concerns in this study. Consequently, our analysis included an assessment of such classifications. Our intention was to be sure that this classification was done consistently according to the instructions given to all participants during the training. We assessed the problem categorisation by checking the software directly in order to confirm the categorisation given by participants to a usability problem. The videos were thoroughly walked through in order to confirm this categorisation.

The tests were conducted on different software systems. There is no joint list of usability problems. In our analyses this is the reason we compared the differences between both conditions using average and standard deviations calculated separately for each condition.

Using the test logs, we analysed the time spent on all the tests. We considered individual and group time consumption. We calculated totals, averages, and percentages to facilitate the analysis. We included in this process all the activities made by all members of the teams in the preparation for the test (e.g., usability plan, usability tasks, etc.) and conducting the test itself. In the analysis, we also considered other participants, such as the users and observers, in order to consider a more realistic context.

Finally, in order to identify significant differences in the data collected, we used independent-sample t tests.

### G. *Overcoming the limitations of the field study*

Wynekoop and Conger [48] classified the field study as a natural setting method normally used for studying current practice and for evaluating new practices. As with any research method, the field study has strengths and weaknesses. The main strengths are that they are practical and in realistic settings.

Braa and Vidgen [49] argue that the field study method is an extension of lab experiments conducted in the particular context of an organisation, and this is something that implies less methodological rigor but conduction in a more realistic environment. The realistic settings used in a field study are useful in terms of exploring a specific phenomenon in conditions close to reality. For example, observation of users' natural behaviours in their own environments was highlighted by Nielsen [50] as an important method used in HCI research.

The main weaknesses of the field study are the difficulties in finding an adequate setting and the control and management of the study.

Considering that having an adequate environment is a key aspect of the design of the field study, the difficulty in finding such an environment becomes a relevant weakness [9]. Real environments may limit the research process (e.g., time restrictions, resource limitations, motivation of participants, etc.). Another weakness is the complexity of the control and management process [51]. The particular characteristics of the field study (i.e., made outside of controlled conditions existing in a lab) make the process complex. For example, a variety of logistics must be considered in the experimental design, as well as the particular conditions presented in the place where the study will be conducted. The management process, among other things, of the data collection is also complex and demands additional efforts considering that the study setting is not necessarily preconditioned to allow the conduction of regular experiments.

Our field study aimed to compare several usability evaluation methods in order to explore how practical and cost effective the methods were.

Our study had the following cited problems: finding adequate environments and controlling and managing the process.

To overcome the difficulty of finding an adequate environment, we did the following. First, we defined a set of conditions that potential organisations and participants had to meet. Second, once we identified potential actors, we randomly selected the number of organisations and participants needed for the study. Finally, using a random distribution, we grouped the actors into different conditions.

To overcome the problem of control and management, we did the following. First, we defined several guidelines to orient the conduction of the study. Second, we provided formal training to the study participants. Third, we provided personalised advice to the participants using alternative channels (i.e., in person, email, chat, and phone). Finally, all data collections were backed up using different alternatives (e.g., CD-ROM copies, public file hosting services, public video-sharing websites, etc.). Although all these measures were taken, it is a fact that the public nature of some tools used to back up the data collection (i.e., the hosting services and video-sharing websites), involves a certain level of risk for such data collections.

## IV. RESULTS

The results section is organised into problems identified by type, task completion time, and time spent on the tests.

### A. *Problems identified by type*

Table III shows an overview of the usability problems identified under the two conditions. The problems are classified by type. The largest number of problems was critical. The lowest number of problems identified was in the category of cosmetic problems. The distribution of all types of problems between the two conditions was relatively uniform.

TABLE III. PROBLEMS IDENTIFIED PER TYPE OF PROBLEM. (%)= PERCENTAGE PER CONDITION.

| Cond.-><br>Problems | L | R |
|---|---|---|
| Critical | 36 (52%) | 33 (56%) |
| Serious | 29 (42%) | 22 (37%) |
| Cosmetic | 4 (6%) | 4 (7%) |
| Total | 69 | 59 |

An independent-sample t test for the number of usability problems identified for the three categories, under both conditions, showed no significant difference (p=0.404). The fact that there are no significant differences between the L and R conditions is a reflection of the similarity of the effectiveness of these methods in terms of the number of problems identified.

### B. *Task completion time*

The task completion time was less in the tests completed under the L condition. In these tests, the users spent a total of 87.6 minutes completing the five tasks assigned to each one. The average time per user/task was 1.94 (SD=0.5). The average task completion time per usability problem identified under the L condition was 1.26. In the tests completed under the R condition, task completion time was 137.4, the average time per user/task was 3.10 (SD=1.3), and the average task completion time per problem was 2.32. In Table IV, we present these results.

An independent-sample t test for the task completion time of the nine users considered under the two conditions showed a significant difference (p=0.018).

The analysis of the videos recorded during the tests completed under the R condition showed delays due to technical problems–mainly in the communication between the actors (i.e., users, test monitor, technician, etc.). In addition, in general, the users in their normal jobs were more distracted. On the contrary, in the case of the tests completed at the laboratory, the users were more focused, and the guidance of the test monitors was more effective.

### C. *Time spent on the tests*

The time spent to complete the tests presents an entirely different perspective to that shown in the previous section. Here, the tests conducted under the R condition consumed less time than that conducted under the L condition.

In Table V, we presented an overview of the time spent in the tests conducted under the two conditions. This table includes the average number of minutes spent on test activities. The standard deviation is shown between parentheses. At the end, the table also shows the average of time per problem in minutes.

These results included all the actors involved in the tests (i.e., users, test monitor, logger, observers, etc.). In this sense, it is possible to consider these results to be more realistic; here, all of the elements/persons required to perform the tests are included. An independent-sample t test, for the average time spent in the tests, for both conditions, showed an extremely significant difference (p<0.001).

TABLE IV. USERS' TASKS COMPLETION TIME AND TIME PER PROBLEM. UP= TOTAL NUMBER OF USABILITY PROBLEMS IDENTIFED PER CONDITION.

| Condition-><br>Test–User | L<br>(UP 69) | | R<br>(UP 59) | |
|---|---|---|---|---|
| | Tot. Minutes | Avg. per task (SD) | Tot. Minutes | Avg. per task (SD) |
| T1–U1 | 10.8 | 2.2 (1.9) | 30.0 | 6.0 (1.3) |
| T1–U2 | 9.7 | 1.9 (1.0) | 18.3 | 3.7 (1.6) |
| T1–U3 | 12.8 | 2.6 (2.5) | 18.7 | 3.7 (1.6) |
| T2–U1 | 6.1 | 1.2 (0.4) | 17.6 | 3.5 (1.8) |
| T2–U2 | 14.3 | 2.9 (0.8) | 13.3 | 2.7 (1.3) |
| T2–U3 | 8.4 | 1.7 (0.7) | 8.9 | 1.8 (0.7) |
| T3–U1 | 7.4 | 1.5 (1.0) | 11.2 | 2.2 (2.4) |
| T3–U2 | 6.9 | 1.4 (0.9) | 9.0 | 1.8 (1.4) |
| T3–U3 | 11.1 | 2.2 (1.1) | 10.5 | 2.1 (2.1) |
| Total<br>Avg. por task<br>(SD) | 87.6<br>1.94<br>(0.5) | | 137.4<br>3.10<br>(1.3) | |
| Avg. task completion time per problem, in minutes | 1.26 | | 2.32 | |

The time spent on each activity during the tests confirms these extremely significant differences for all of the activities–except in the analysis. In preparation, conducting the tests, and moving staff, the independent-sample t tests for the time spent in the three tests conducted under each condition, showed extremely significant differences (p<0.001 for all of the cases). In the case of the analysis, the difference was significant (P=0.045).

## V. DISCUSSION

The discussion section is organised into two parts. First, we will reflect on the effectiveness of RUT to overcome the cost obstacle. Next, we reflect on how RUT helps to handle the practical constraints previously presented in the related works section.

TABLE V. TIME SPENT IN THE TESTS. UP= TOTAL NUMBER OF USABILITY PROBLEMS IDENTIFIED PER CONDITION .

| Condition-><br>Activity | L<br>(UP 69) | R<br>(UP 59) |
|---|---|---|
| Preparation | 2500 (102) | 1580 (123) |
| Conducting test | 1320 (73) | 840 (42) |
| Analysis | 980 (157) | 710 (71) |
| Moving staff/users | 1110 (107) | 160 (57) |
| Tot.time spent per test | 5910 (220.5) | 3290 (102) |
| Avg. time per problem in minutes | 85.7 | 55.8 |

### A. *Overcoming the cost obstacle*

Usability evaluations made using the remote synchronous testing method are a cost-effective alternative to integrating usability evaluations into software projects. The number of usability problems identified by this method is similar to that obtained by conventional tests made in a usability laboratory. Additionally, there is a significant difference between the time spent on the remote synchronous test method and that spent on the tests made in the lab.

We confirmed the feasibility of conducting usability evaluations by software developers using diverse methods, including the remote synchronous testing method [24][28][41]. In parallel to this practical feasibility, our study also proved the economic feasibility of the remote synchronous testing technique by taking economic advantage of consideration of the developers' conduction of usability tests as was suggested by Bruun and Stage [28], and Skov and Stage [24]. Using developers to conduct usability evaluations, it was not necessary to hire external independent usability experts, thus reducing the cost of the process as suggested by Bruun [52].

In addition, we also confirmed the similarity to the number of problems identified by the conventional lab method [33]. However, in the case of the time spent, our results differ from those of others [33] who argue that the time spent to conduct tests using lab and remote synchronous tests was quite similar. In our case, the difference in time consumption for both methods was significantly favourable in the remote synchronous testing method. A detailed analysis of the test logs showed us that, in the tests made under the L condition, the logistic matters consumed much more time than in the tests under the R condition. Considering our aim of confirming previous findings in a realistic development context, logistic matters must be considered as factual components of any usability test.

The analysis of the procedures followed the conducting of the tests (reported in the usability problem reports) and the test logs showed that when using the remote synchronous testing method, it is possible to achieve several practical advantages that save time in the tests.

It is possible to contextualise these advantages in the results of the time spent on test activities shown in Table V. First, in the case of the preparation activities, the virtualisation of the complete coordination process saved time and effort. The coordination between teams and other actors was easier and more efficient using email, chat, videoconferences, etc.

Second, in the activities of conducting the tests, it was easy and efficient to use all the software tools used during the tests. Even when considering that the task completion time was shown to be better in the tests made under the L condition (see Table IV), differences in the overall process were evident due to this task completion time only being related to the time spent by users to complete the tasks. On the contrary, in the conducting activities of the tests, all of the elements and actors required to conduct the whole test are included (i.e., users, test monitor, logger, observers, etc.)

Third, the difference in the analysis was also significant due to the technological tools that facilitated the conducting of the analysis sessions by the facilitator. In a certain way, the videos also showed that the virtualisation of the process seems to produce a shared feeling about the relevance of productivity during the virtual sessions.

Finally, the results in the moving activities explain themselves. In the realistic development context used in this study, it is clear that avoiding the movement of the usability evaluation staff is one of the most relevant advantages in terms of time consumption.

In general, all of the advantages of the remote synchronous test cited in literature were confirmed in the realistic contexts considered in our study [33][47]. In the case of the disadvantages, we could only identify–in the analysis of the test logs–some problems in the setting of the hardware and software tools used in the process [33].

At this point in the discussion, the economic advantages of the remote synchronous testing method had become evident. Furthermore, this method also helps to handle other practical problems of the integration of usability evaluations into software projects.

### B. *Overcoming practical obstacles*

In our study, we have also confirmed the feasibility of the active participation of software developers in usability evaluations [5][24][28]. The participants played several roles in the usability evaluation teams (e.g., test monitor, logger, observer, and technician). This confirmation is relevant when considering the context used in our study (i.e., lab and remote synchronous tests under more realistic conditions). The design of our experiment proved to be very useful because all of the teams actively participated in all of the processes (i.e., planning and conducting the test) and with impartiality. It is a fact that these levels of participation of developers in usability evaluations may positively impact their perspective regarding usability and the HCI practitioners [27] and will reduce the tensions between SE and HCI practitioners [19][25].

Furthermore, in the case of the problem related to the lack of a formal application of HCI techniques, our experiment found that using guidelines and basic training, it is possible to prepare developers for conducting usability evaluations. In a certain way, the theory used to inspire the guidelines used in the tests has followed the suggested approach [16] of using standards to help the integration of usability evaluation in software projects. The analysis of the dynamic of the tests registered in the videos did not show any particular significant problems.

In the case of the tests made using the remote synchronous testing method, the guidelines were fundamental in conducting the remote process. Considering the similarity of the results in the remote synchronous tests and those obtained in the lab, it is clear that the guidelines served their purpose.

Considering these facts, we can conclude that using guidelines based on standards, it is possible to improve the perception of the lack of appropriate methods for usability evaluation [20][22].

Finally, our study also found that the reported problem [20][22][30] relating to the participation of customers and users could be handled well using the remote synchronous testing method. The users do not need to drastically change their activities. Certainly, the task completion time was higher in the remote synchronous testing method however, putting this element in perspective for the entire process; it is always possible to see the strengths of the remote synchronous testing method. Furthermore, other actors did not have to go to the lab.

## VI. Conclussion and Future Work

In this paper, we presented the results of a study aimed at comparing the remote synchronous test method against the classical laboratory-based think-aloud method in a realistic software development context. Final-year students who had 18 months of practical experience conducted several tests. Although, the tests were made on software systems for different organisations and purposes, the scope of these software systems was carefully controlled in order to provide similar settings for the study.

Our study confirmed that remote synchronous testing is a practical and cost-effective alternative for integrating usability evaluations into software projects. The study has shown that there is no statistical significant difference in the number of problems identified using the remote synchronous test method when compared to the usability lab. Even considering that the task completion time in the lab was 37% quicker, the time spent to complete all of the remote synchronous tests was 44% quicker. The statistical analysis has shown that the difference was extremely significant. These results included all the actors involved in the tests (i.e., users, test monitor, logger, observers, etc.), which implies a more real context in terms of the whole testing process. In this case, the field study has shown that with the remote synchronous test it is possible to reduce the ‘actual cost obstacle’ in order to allow economical conduction of usability evaluations.

The identification of a similar number of usability problems and lower time consumption make remote synchronous test method a good alternative. Using this method, it is possible to involve more software developers in the conduction of usability testing. Such an aim only requires basic training, guidelines, and essential advice. Basic guidelines and training allow handling the problems related to the methods. Finally, one of the most relevant advantages of this method is to facilitate the participation of users, developers, and other potential actors in the tests. Avoiding unnecessary movements allows their participation to be easily justified.

In our study, we were focused on the problems identified and the time consumption metrics in a realistic development context. For future work, it is suggested that for the same context a deeper analysis of other metrics, such as the improvement of the perspective of software developers regarding usability, which is another expected result of close participation of developers in usability evaluations, should be conducted.

## VII. Limitations of the Study

Our study has two limitations. First, the participants in the study were final-year undergraduate students. Nevertheless, the real conditions present in our study have allowed for a control of this bias. In addition, we think that it is possible to consider these advanced students as novice software developers because they share similar characteristics. We base such a statement on three main facts. First, Bruun and Stage [28] defined novice developers as persons with limited job experience related to usability engineering and no formal training in usability engineering methods. In this sense, advanced students share similar characteristics to the novice developers. Second, the students mainly conducted usability evaluations in the PhD project. To perform these activities, students had to use several soft skills (e.g., defining user tasks, documenting results, following a method, working with real users, working in teams, etc.). According to Begel and Simon [53], novice developers (as well as the students who participated in my research), usually have serious constraints when it comes to these soft skills because these issues are normally less well supported in university pedagogy. Finally, as with novice developers, the students who participated in the PhD project were not preconditioned with extensive previous work experience.

Second, we used only two usability evaluation techniques. However, our selection considered an ideal benchmark of high interaction with users (lab) and the alternative option, which was the focus of our study.

## Acknowledgment

The research behind this paper was partly financed by UNA, MICIT, CONICIT (Costa Rica), and the Danish Research Councils (grant number 09-065143). We are very grateful to the participants, observers, facilitators, organisations, and users that helped us in this research.

## References

[1] F. Lizano and J. Stage, “Usability Evaluations for Everybody, Everywhere: A field study on Remote Synchronous Testing in Realistic Development Contexts,” Proc. ICDS 2014, pp. 74-79. 2014.

[2] Y. Jia, “Examining Usability Activities in Scrum Projects–A Survey Study,” Doctoral dissertation, Uppsala Univ., 2012.

[3] G. Lindgaard and J. Chattratichart, “Usability testing: what have we overlooked?,” Proc. SIGCHI, ACM Press, pp. 1415-1424, 2007.

[4] P. Bourque and R. Fairley, “SWEBOK : Guide to the Software Engineering Body of Knowledge Version 3.0.,” IEEE Computer Society, 2014.

[5] R.T. Hoegh, C.M. Nielsen, M. Overgaard, M.B. Pedersen, and J. Stage, “The impact of usability reports and user test observations on developers' understanding of usability data: An exploratory study,” in International journal of Human-Computer Interaction, 21(2), pp. 173-196, 2006.

[6] R.H. Rasch and H.L. Tosi, “Factors Affecting Software Developers' Performance: An Integrated Approach,” MIS quarterly, vol.16(3), 1992.

[7] G. Hertel, S. Niedner, and S. Herrmann, “Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel,” Research policy, vol. 32(7), pp. 1159-1177, 2003.

[8] J. Nielsen, Usability engineering, Morgan Kaufmann Publishers, 1993.

[9] A. Abran, J.W. Moore, P. Bourque, R. Dupuis, and L.L. Tripp, “Guide to the Software Engineering Body of Knowledge: 2004 Edition-SWEBOK,” IEEE Computer Society, 2004.

[10] N. Juristo and X. Ferre, "How to integrate usability into the software development process," Proc. the 28th international conference on Software engineering, ACM Press, pp. 1079-1080, 2006.

[11] K. Radle and S. Young, "Partnering usability with development: How three organizations succeeded," IEEE Software, vol.18(1), pp.38-45, 2001.

[12] T. Granollers, J. Lorés, and F. Perdrix, "Usability engineering process model. Integration with software engineering," Proc. HCI International, pp 965-969, 2003.

[13] P.F. Drucker, "Knowledge-Worker Productivity: The Biggest Challenge," in California management review,vol.41(2), pp.79-94, 1999.

[14] A. Hernandez-Lopez, R. Colomo-Palacios, and A. Garcia-Crespo, "Productivity in software engineering: A study of its meanings for practitioners: Understanding the concept under their standpoint," Proc. Information Systems and Technologies (CISTI), IEEE Press, pp. 1-6, 2012.

[15] G.H. Meiselwitz, B. Wentz, and J. Lazar, Universal Usability: Past, Present, and Future, Now Publishers Inc., 2010.

[16] H. Fischer, "Integrating usability engineering in the software development lifecycle based on international standards," Proc. SIGCHI symposium on Engineering interactive computing systems, ACM Press, pp. 321-324, 2012.

[17] D. Nichols and M. Twidale. *The usability of open source software*. [Online]. Available from: http://firstmonday.org/ojs/index.php/fm/article/view/1018/939 2014.11.14

[18] A. Seffah, M.C. Desmarais, and E. Metzker, "HCI, Usability and Software Engineering Integration: Present and Future," In Human-Centered Software Engineering—Integrating Usability in the Software Development Lifecycle, Springer: Berlin, Germany, pp. 37-57, 2005.

[19] O. Sohaib and K. Khan, "Integrating usability engineering and agile software development: A literature review," Proc. ICCDA, IEEE Press, vol. 2, pp. V2-32, 2010.

[20] C. Ardito, P. Buono, D. Caivano, M.F. Costabile, R. Lanzilotti, A. Bruun, and J. Stage, "Usability evaluation: a survey of software development organizations," Proc. SEKE, pp. 282-287, 2011.

[21] J. Gulliksen, I. Boivie, J. Persson, A. Hektor, and L. Herulf, "Making a difference: a survey of the usability profession in Sweden," Proc. NordiCHI, ACM press, pp. 207-215, 2004.

[22] F. Lizano, M.M. Sandoval, A. Bruun, and J. Stage, "Usability Evaluation in a Digitally Emerging Country: A Survey Study," Proc. INTERACT, Springer Berlin Heidelberg, pp. 298-305, 2013.

[23] J. Nielsen, "Guerrilla HCI: Using discount usability engineering to penetrate the intimidation barrier," in Cost-justifying usability, pp. 245-272, 1994.

[24] M.B. Skov and J. Stage, "Training software developers and designers to conduct usability evaluations," in Behaviour & Information Technology, 31(4), pp. 425-435, 2012.

[25] J.C. Lee and D.S. McCrickard, "Towards extreme (ly) usable software: Exploring tensions between usability and agile software development," in Proc. Agile Conference (AGILE), IEEE Press, pp. 59-71, 2007.

[26] X. Ferré, N. Juristo, and A. Moreno, "Which, When and How Usability Techniques and Activities Should be Integrated," in Human-Centered Software Engineering - Integrating Usability in the Software Development Lifecycle, Springer Netherlands, pp. 173-200, 2005.

[27] J.C. Lee, "Embracing agile development of usable software systems," In Proc.CHI'06 extended abstracts, ACM Press, pp. 1767-1770, 2006.

[28] A. Bruun and J. Stage, "Training software development practitioners in usability testing: an assessment acceptance and prioritization," Proc. OzCHI, ACM Press, pp.52-60, 2012.

[29] Z. Hussain, M. Lechner, H. Milchrahm, S. Shahzad, W. Slany, M. Umgeher, P. Wolkerstorfer, "Practical Usability in XP Software Development Processes," in Proc. ACHI, pp. 208-217, 2012.

[30] J.O. Bak, K. Nguten, P. Risgaard, and J. Stage, "Obstacles to Usability Evaluation in Practice: A Survey of Software Development Organizations," Proc. NordiCHI, ACM Press, pp.23-32, 2008.

[31] V. Bellotti, "Implications of current design practice for the use of HCI techniques," Proc. the Fourth Conference of the British Computer Society on People and computers, Cambridge University Press, pp. 13-34, 1988.

[32] K. Ehrlich and J. Rohn, "Cost justification of usability engineering: A vendor's perspective," in Cost-justifying usability, 1994.

[33] M.S. Andreasen, H.V. Nielsen, S.O. Schrøder, and J. Stage, "What happened to remote usability testing?: an empirical study of three methods," Proc. SIGCHI, ACM Press, pp. 1405-1414, 2007.

[34] A. Bruun, P. Gull, L. Hofmeister, and J. Stage, "Let your users do the testing: a comparison of three remote asynchronous usability testing methods," Proc. SIGCHI, ACM Press, pp. 1619-1628, 2009.

[35] R. Jeffries, J.R. Miller, C. Wharton, and K. Uyeda, "User interface evaluation in the real world: a comparison of four techniques," Proc. SIGCHI, ACM Press, pp. 119-124, 1991.

[36] S. Alshaali, Human-computer interaction: lessons from theory and practice. Doctoral dissertation, University of Southampton, 2011.

[37] A. Holzinger, "Usability engineering methods for software developers," Communications of the ACM, vol.48(1), pp.71-74, 2005.

[38] J. Kjeldskov, M.B. Skov, and J. Stage, "Instant data analysis: conducting usability evaluations in a day," Proc. NordiCHI, ACM Press, pp. 233-240, 2004.

[39] T. Borgholm and K.H. Madsen, "Cooperative usability practices," Communications of the ACM, vol.42(5), pp. 91-97, 1999.

[40] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," Proc. SIGCHI, ACM Press, pp. 249-256, 1990.

[41] H.R. Hartson, J.C. Castillo, J. Kelso, and W.C. Neale, "Remote evaluation: The network as an extension of the usability laboratory," Proc. CHI, ACM Press, pp. 228-235, 1996.

[42] S. Dray and D. Siegel, "Remote possibilities?: international usability testing at a distance,", Interactions, vol.11(2), pp. 10-17, 2004.

[43] F. Menghini. *Remote usability testing*. [Online]. Available from: http://internotredici.com/article/remoteusabilitytesting/ 2014.11.14

[44] F. Paternò, "Models for universal usability," Proc. the 15th French-speaking conference on human-computer interaction on 15eme Conference Francophone sur l'Interaction Homme-Machine, ACM Press, pp. 9-16, 2003.

[45] M. Hammontree, P. Weiler, and N. Nayak, "Remote usability testing," in Interactions, 1, 3, pp. 21-25, 1994.

[46] K.E. Thompson, E.P. Rozanski, and A.R. Haake, "Here, there, anywhere: Remote usability testing that works," Proc. Conference on Information Technology Education, ACM Press, pp. 132–137, 2004.

[47] J. Rubin and D. Chisnell, Handbook of usability testing: how to plan, design and conduct effective tests, John Wiley & Sons, 2008.

[48] J.L. Wynekoop, S.A. Conger, "A review of computer aided software engineering research methods," Department of Statistics and Computer Information Systems, School of Business and Public Administration, Bernard M. Baruch College of the City University of New York, 1992.

[49] K. Braa and R., Vidgen, "Interpretation, intervention, and reduction in the organizational laboratory: a framework for in-context information system research,", Accounting, Management and Information Technologies, vol.9(1), pp.25-47, 1999.

[50] J. Nielsen. *Best Application Designs.* [Online]. Available from: http://www.nngroup.com/articles/best-application-designs/ 2014.11.14

[51] J. Kjeldskov and C. Graham, "A review of mobile HCI research methods," Human-computer interaction with mobile devices and services, Springer Berlin Heidelberg, pp. 317-335, 2003.

[52] A. Bruun, Developer Driven and User Driven Usability Evaluations. Doctoral dissertation, Videnbasen for Aalborg Universitet, Det Teknisk-Naturvidenskabelige Fakultet, Aalborg University, The Faculty of Engineering and Science, 2013.

[53] A. Begel and B. Simon, "Novice software developers, all over again," Proc the Fourth international Workshop on Computing Education Research, ACM Press, pp. 3-14, 2008.