Usability Evaluations for Everybody, Everywhere:

A field study on Remote Synchronous Testing in Realistic Development Contexts

Fulvio Lizano
Department of Computer Science
Aalborg University
Aalborg, Denmark
fulvio@cs.aau.dk

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

***Abstract*—Although Human-Computer Interaction (HCI) 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, for instance, the remote synchronous testing method. This paper presents results from a field study in which we compared 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. The results show 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, by 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 positively impact that remote synchronous testing could have in the digital accessibility of the software, by allowing extensive use of usability evaluation practices into software development projects.**

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

I. INTRODUCTION

Usability has a significant impact on software development projects [15]. Common usability activities, as usability evaluations, are relevant and strategic in diverse contexts (e.g., organizations, software development process, software developers and users) [3], [13].

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 [5], [12] and is something that makes it very difficult to justify some HCI activities [20].

Bearing this in mind, any effort to integrate usability evaluations into software projects must necessarily consider practical and cost-effective methods, such as the remote synchronous test.

In this paper, 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 summarized, the paper presents the analysis before concluding with suggestions for future work.

II. RELATED WORKS

Integration efforts of usability evaluations into software projects have economic and practical constraints.

High consumption of resources in usability evaluations is a recurrent perception in diverse contexts [2], [3], [19], [22], [23]. This fact could explain why usability has a lower valuation for the organization's top management [8], becoming manifest by the lack of respect and support for usability and the HCI practitioners [9]. Therefore, cost-justification of usability may be difficult for many companies when it is perceived as an extra cost or feature [20].

On the other hand, three of the most cited practical constraints are related to: the difference of perspectives between HCI and Software Engineering (SE) practitioners, the absence or diversity of methods and, finally, the users' participation.

The first constraint related to the difference of perspectives between HCI and SE practitioners is contextualized in the difference of opinions they have about what is important in software development [17]. This diversity of perspectives results in contradictory points of view regarding how usability testing should be conducted and is something that 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 centered on how the software should be developed in a practical an economical way [27]. These conflicting perspectives result in tensions between software developers and HCI practitioners [18], [27].

The second constraint relates to the absence or diversity of methods, and has two opposing views. Firstly, some researchers report a lack of appropriate methods for usability evaluation [2], [19] or a lack of formal application of HCI and SE methods [15]. This situation may explain why the UCD community has expressed criticism about the real application of some software development principles [25]. Secondly, it is reported that the existence of numerous and varied techniques and methodologies in the HCI and SE fields could hamper the integration [18].

Finally, the participation of customers and users has become another relevant limitation for the integration of usability evaluations into software projects [2], [3], [19]. 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. Firstly, in the case of the difference of perspectives between HCI and SE practitioners, some studies have suggested that increased participation of developers in usability testing could positively impact their valuation of usability [13]. This improvement in the developers' perspectives could make them more conscious of the relevance of HCI techniques.

Secondly, with respect to the absence or diversity of methods, an integration approach based on international standards is proposed [7] in order to enable consistency, repeatability of process, independence of organizations, quality, etc. A similar approach suggests the integration of HCI activities into software projects by using SE terminology for HCI activities [6].

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 [14].

These aforementioned studies were conducted on limited realistic contexts, e.g., literature reviews [7], [20], [23], [25], [27], surveys [2], [5,], [9], [15], [19], experiments in labs [22], [26] and case studies [13], [18]. Other papers cited above present proposals of projects or methods [6], [8], [17]. There are only three studies with a more empirical base in more realistic contexts [4], [13], [14]. Confidence in the results of these studies should be improved by other studies made in a realistic development context.

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).

By using remote synchronous testing, the test is conducted in real time, but the evaluators are separated spatially from the users [1]. The interaction between the evaluators and the users is similar to those at a usability lab. There are many studies that confirm the feasibility of remote usability testing methods [1], [10], [28]. Actually, there is a clear consensus regarding the benefits obtained by 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) [1], [24]. The main disadvantages are related to problems of generating enough trust between the test monitor and users, a longer setup time, and difficulties in re-establishing the test environment if there is a problem with the hardware or software [1].

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.

All of these teams were formed by final-year students of SE who had 18 months of practical experience working in software development. This experience is the result of an academic project created by the students by developing a software system in a real organization.

A. 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 pre-selected as potential participants in the experiment. Finally, we randomly selected six teams who were randomly distributed throughout the R and L conditions.

The teams were formed by final-year students who were finishing their last course in System Engineering. These participants were organized 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 of practical academic activity by developing a software system in a real organization that sponsored the project. These organizations 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 1–10 (SD=0.33). As an incentive for participation, the participants received extra credits. The conditions, code, members and software are presented in Table I.

B. 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 [16], [24]. In addition, a workshop was made in order to putting into practice the contents of the training materials. The participants received specific instructions in order to consider three categories of usability problems: critical, serious, and cosmetic [1]. The number of hours spent in training was 10 (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.

TABLE I. TEAMS, MEMBERS, AND STAFF FOR THE USABILITY EVALUATION

Cond.	Code	Members	Software
L	L1	3 males	Students' records in a private 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 center 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

C. Procedure

The design of the experiment increased confidence in the results and 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 10 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 thought 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 [16].

D. Settings

The test conducted under the L condition used a state-of-the-art usability lab and think-aloud protocol [21], [24]. Each test included three sessions where the users were sat in front of the computer and the test monitor was sat next the users. The logger and observers were present in the same room. In the case of the R condition, the tests were based on the remote synchronous testing [1]. All participants were spatially separated. Users were in the sponsors' facilities. Each test included three sessions with users.

E. Data collection and analysis

Each user session was video recorded. The video included the software session recorder (video capture of screen) and a small video image of the user. Under R conditions, the video also recorded the image of the test monitor. 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.

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

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

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 classification. 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 categorization by checking the software directly in order to confirm the categorization given by participants to a usability problem. The videos were thoroughly walked through in order to confirm this categorization.

The tests were conducted on different software systems. There is not a joint list of usability problems. This is the reason why, in our analysis, we compared the differences between both conditions by using average and standard deviations calculated separately for each condition.

Using the test logs, we analyzed the time spent in 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 of the test (e.g., usability plan, usability tasks, etc.) and the conducting of 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.

IV. RESULTS

A. Problems identified per type

Table II shows an overview of the usability problems identified under the two conditions. The problems are classified by their 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, among the two conditions, was relatively uniform. 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

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

Condition-> Test–User	**L (UP 69)**		**R (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 Avg. por task (SD)	87.6 1.94 (0.5)		137.4 3.10 (1.3)	
Avg. task completion time per problem, in minutes	1.26		2.32	

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 made 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 made under the R condition, the 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 III, 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 made 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 made at the laboratory, the users were more focused, and the guidance of the test monitors was more effective.

C. *Time spent in 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.

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

Condition-> Activity	**L (UP 69)**	**R (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

In Table IV, 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 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 ($p<0.001$).

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

Usability evaluations made by 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 [4], [11], [26]. In addition, we also confirmed the similarity to the number of problems identified by the conventional lab method [1]. However, in the case of the time spent, our results differ from those of others [1] who argue that the time spent to conduct tests by using lab and remote synchronous tests was quite similar. In our case, the difference in time consumption for both methods was significantly favorable 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, by using the remote synchronous testing method, it is possible to achieve several practical advantages that save time in the tests.

It is possible to contextualize these advantages in the results of the time spent on the tests' activities shown in Table IV. Firstly, in the case of the preparation activities, the virtualization of the complete coordination process saved time and effort. The coordination between teams and other actors was easier and more efficient by using email, chat, video conferences, etc.

Secondly, in the activities of conducting the tests it was also 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 III), 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.)

Thirdly, 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 virtualization 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 the literature were confirmed in the realistic contexts considered in our study [1], [24]. 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 [1].

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

In our study, we have also confirmed the feasibility of the active participation of software developers in usability evaluations [4], [13], [26]. 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 process (i.e., planning and conducting of the test) and with impartiality. It is a fact that these levels of participation of developers in usability evaluations may impact positively upon their perspective regarding usability and the HCI practitioners [17] and will reduce the tensions between SE and HCI practitioners [18], [27].

Furthermore, in the case of the problem related to the lack of formal application of HCI techniques, our experiment found that by 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 [7] of using standards to help the integration of usability evaluation into 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 by 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, by using guidelines based on standards, it is possible to improve the perception of the lack of appropriate methods for usability evaluation [2], [19].

Finally, our study also found that the reported problem [2], [3], [19] relating to the participation of customers and users can be handled well by 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 but, putting this element in perspective for the whole 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

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

The identification of a similar number of usability problems and lower time consumption, make of Remote Synchronous a good alternative for integrating usability evaluations into software projects. By using this method it is possible to involve more software developers into the conduction of usability testing. Such aim only requires basic training, guidelines and essential advice. Basic guidelines and training allows 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. By avoiding unnecessary movements of these persons, their participation will be easily justified

Our study has two main limitations. Firstly, 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. Secondly, 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. 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.

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, organizations and users that helped us in this research.

REFERENCES

[1] 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, 2007, pp. 1405-1414.

[2] C. Ardito et al., “Usability Evaluation: a survey of software development organizations,” Proc. 33 International Conference on Software Engineering & Knowledge Engineering, 2011. Pp. 282-287.

[3] 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, 2008, pp.23-32.

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

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

[6] 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, 2005, pp. 173-200.

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

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

[9] 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, 2004, pp. 207-215.

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

[11] 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, 1996, pp. 228-235.

[12] 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, June 2012, pp. 1-6.

[13] 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), 2006, pp. 173-196.

[14] Z. Hussain et al., “Practical Usability in XP Software Development Processes,” in Proc. ACHI, January 2012, pp. 208-217.

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

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

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

[18] 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, August 2007, pp. 59-71.

[19] 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, 2013, pp. 298-305.

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

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

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

[23] D. Nichols and M. Twidale, “The usability of open source software,” in First Monday, 8(1), [online], Available: http://firstmonday.org/ojs/index.php/fm/article/view/1018/939 , [retrieved: 01, 2014], 2003

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

[25] A. Seffah, M.C. Desmarais, and E. Metzker, “HCI, Usability and Software Engineering Integration: Present and Future,” In Human-Centered Software Engineering, Seffah, A. et al. (eds.), Springer: Berlin, Germany, 2005.

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

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

[28] 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, 2004, pp. 132–137.