

How do software engineers apply an early usability inspection technique? A qualitative study

Natasha Malveira C. Valentim, Tayana Conte
USES Research Group, Universidade Federal do Amazonas
Manaus, Brazil
{natashavalentim, tayana}@icomp.ufam.edu.br

Bernardo Estácio, Rafael Prikkladnicki
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre, Brazil
{bernardo.estacio, rafaelp}@puccrs.br

Abstract — Usability inspections can be employed in early phases of the software development process. They improve usability through artifacts that are built during the development of the software. These artifacts will influence the usability of the developed software. Usability inspection techniques have been proposed and considered as an effective alternative for addressing usability issues in early phases. However, these techniques are often avoided by software engineers due to their lack of experience and knowledge in the field. Therefore, there is an opportunity to investigate how industry practitioners have employed an early usability inspection technique in practice. This paper describes an observational study in the industry aimed at eliciting the process used by software engineers when applying an early usability inspection technique. We analyzed the qualitative data, discussing their impact in the improvement of the technique. The results indicated which steps the software engineers adopted in the technique's application.

Keywords- Usability evaluation, usability inspection; early usability; qualitative study.

I. INTRODUCTION

Usability is universally acknowledged as a significant aspect of the overall quality of interactive systems [1]. Including usability allows benefits such as improving user productivity and reducing training and documentation costs [2]. Therefore, a large number of researchers have investigated ways to include usability in software development [3], [4]. Donahue [5] says that investments in usability have allowed benefits such as income increase. This has motivated more organizations to consider usability as a relevant factor in their software products [6].

However, Seffah and Metzker [7] highlight some of the following challenges when including usability into the development process: (a) usability activities are usually separated from the software development process, and (b) the notations and tools in which usability is considered are different from those employed in the development process. Furthermore, the development processes do not take advantage of the intermediate artifacts that are produced during early stages (i.e., requirements and design stages). These intermediate artifacts (e.g., navigational models) are mainly employed to guide software engineers and to document the application. Since the traceability between these artifacts and the final application is not well defined, performing evaluations using these artifacts can be difficult [8].

For this reason, it is important to propose technologies that can be applied by software engineers in the usability evaluation of the artifacts that are employed in the early stages of the development process. The benefits of using this type of technologies are: (i) to assist developers in learning about usability and interaction design and (ii) to reduce the usability evaluation costs because often these evaluations are performed only when the software is ready, generating rework and increasing costs with repairs and improvements.

In this context, this paper describes an observational study with 15 software engineers who applied a usability inspection technique in mockups. They had between 1 and 13 years of experience in the development of projects with the software industry, both Web applications and Mobile applications. In this observational study, we intended to 'look inside' the inspection process, enabling us to understand how software engineers apply an early usability inspection technique.

The early usability inspection technique analyzed in this study is called MIT 2. This technique aims at evaluating the usability through mockups [9]. The MIT 2 is part of a set of techniques called Model Inspection Techniques for Usability Evaluation (MITs), composed by two other techniques: MIT 1 (for the usability evaluation of use case specifications) and MIT 3 (for the usability evaluation of activity diagrams). The MITs intend to reduce the cost of fixing usability problems in artifacts that are employed in the early stages of the development process. They have verification items that guide the software engineers in the discovery of usability problems. Therefore, investigating how software engineers apply this technique during an early usability inspection is important to understanding such practice.

The remainder of this paper is organized as follows: Section 2 discusses the concept of Early Usability. Section 3 presents the MITs technique. Section 4 describes the planning and the execution of the Observational Study. Sections 5 and 6 present the results of the Quantitative and Qualitative Analysis, respectively. Section 7 presents some discussions. Section 8 describes the threats to validity. Section 9 concludes the paper.

II. EARLY USABILITY

Early Usability considers the usability in early phases of the development lifecycle. The goal of Early Usability is to improve usability through artifacts that are built during the software development that will influence the quality of the developed software. Early Usability can help reduce the

number of problems detected in software development projects. It also provides benefits such as increasing the quality of the develop software and higher user satisfaction [2]. According to Fernandez et al. [8], if usability problems are repaired earlier, the quality of the final application can be improved, saving resources in the development stage. Therefore, contributing to reducing the cost of the development process.

Propp et al. [10] proposed a representative example of a technology that considers Early Usability. This approach focuses specifically in the development of interactive systems based on task models. To evaluate usability, first we use the task model to control the user interaction at a degree of abstraction based on tasks. After having introduced the usage of a task engine for task model based on capture, it is necessary to perform the connection between the initial task model and the further refined software artifacts at different stages of the development. To use this approach, it is necessary to adopt the development process based on the proposed model.

Hornbæk et al. [3] propose the UCE method (Use Case Evaluation) for the usability evaluation based on use cases that employs Nielsen’s [11] heuristics as a basis. This method consists of three activities: (1) Inspection of Use Cases, that seeks to identify usability problems that the evaluator is convinced one or more prospective users will experience, (2) Assessment of Use Cases, that seeks to assess the quality of the use cases, and (3) Documentation of Evaluation, where the results are compiled into a coherent evaluation product. This method does not require computer support itself.

III. MODEL INSPECTION TECHNIQUE FOR USABILITY EVALUATION

The MITs are reading techniques to include usability in the early stages of the development process (analysis and design phase), in order for the final applications to become easier to use. According to Travassos et al. [12], reading techniques are a type of inspection technique that contains a series of steps for the individual analysis of a software product in order to achieve the necessary understanding for a specific task. Thus, the MITs main innovation is the verification items that serve as a guide to interpret Nielsen’s [11] heuristics. That is, the MITs guide software engineers during the usability evaluation of Use Case specifications (MIT 1), Mockups (MIT 2) and Activity Diagrams (MIT 3). This allows software engineers to be assisted by the technique during the search for usability problems, even if they have little experience in usability.

One artifact that is often available in early stages of the software development is the mockup. It is an important artifact for both software development and for the design of user interfaces. According to Luna et al. [13], mockups are artifacts employed to represent aspects of the user interface serving as sketches of the applications. They are intended to be developed quickly to reflect the needs of customers in terms of presentation more significantly than the requirements expressed in written language. Therefore, evaluating the usability of these artifacts allows the discovery of problems early. The MIT 2 technique was built for this purpose: to assist in the discovery of problems in the initial stages of the development process, through mockups, even the early prototypes made of low-fidelity materials. It has verification items that are based on the

heuristics by Nielsen [11], but in a more guided way. That is, the verification items guide the software engineers, even if not usability experts, in the search of usability problems. The current version of MIT 2 is available in a technical report [9]. Table I presents some verification items of MIT 2.

TABLE I. PART OF THE MIT 2 TECHNIQUE [9].

MIT-2AE. Error prevention Heuristic	
Verification Item 2AE3	Verify if there is any system warning that alerts, through messages or informational texts, that what the user is doing may be inappropriate at that time;
Verification Item 2AE4	Verify if all available options, buttons and links have names that clearly define what results or conditions will be met.

The steps for using the MIT 2 technique are shown in Figure 1. These steps are: (1) to evaluate the mockup using the MIT 2 technique and (2) to identify usability problems. In order to illustrate the MIT 2’s steps, we have employed it to evaluate the usability of a mockup. This mockup was created based on a page of the SION System¹. This page is used in the SION System to register a course of a training center. In the next paragraphs we describe how we applied the steps to perform a simple inspection of the mockup from the SION System. This example shows only part of the inspection of the SION System, since we are only evaluating one of its mockups.

The first step for the identification of usability problems is to evaluate the usability through verification items. In other words, software engineers must check if the mockup meets all the usability verification items described within each heuristic. Table I shows an example of the usability verification items.

In order to identify usability problems (2nd step), software engineers must point in the mockup which part did not meet the usability verification items. If we look at Figure 1 and Table I, we can relate the nonconformities of the usability verification items in Table I with the augmented element A in Figure 1.

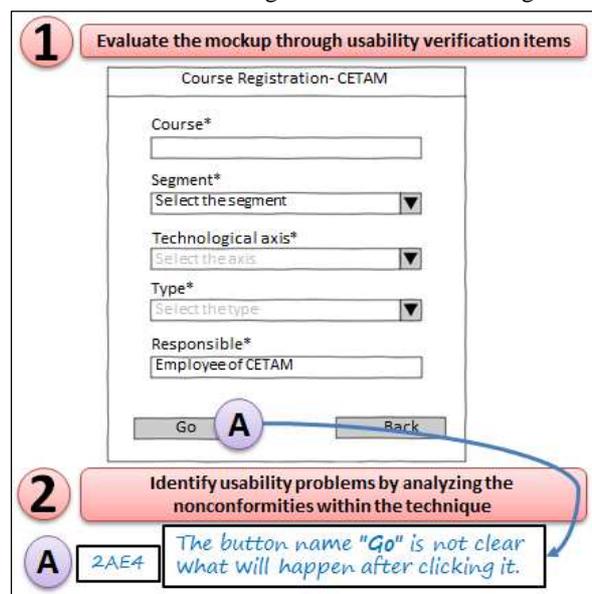


Figure 1. Example of the MIT 2’s steps.

¹ <http://sion.secti.am.gov.br/principal/>.

The 2AE4 verification item requires software engineers to verify if all available options, buttons and links have names that clearly define the results that will be achieved. The name of the button “Go” does not make it clear if the course will be registered or if the user will go forward into the next screen (see Figure 1 element A). In other words, the button name does not make clear what will happen after clicking it.

IV. THE OBSERVATIONAL STUDY

To support the development and validation of the MIT 2, we have adopted the empirical methodology presented in Shull et al. [14]. It comprises four stages: (1) feasibility studies: to determine the usage possibility of the technology; (2) observational studies: to improve the understanding and the cost-effectiveness of the technology; (3) case studies in real lifecycles: to characterize the application of the technology during a real lifecycle; (4) case studies in industry: to identify if the application of the technology fits into industrial settings.

In order to verify the possibility of employing the MIT 2 technique, the authors conducted two feasibility studies. Valentim et al. [15] and Valentim and Conte [16] describe the results of the studies. The statistical test results showed that MIT 2 obtained similar effectiveness and efficiency as the Heuristic Evaluation in both the first and the second feasibility study. This indicates that further studies need to be performed to identify which part of the inspection process with the MIT 2 needs improvement. We expect that software engineers can use MIT 2 to ensure the quality of their mockups. To achieve this goal, we carried out the second stage of the methodology by obtaining a detailed understanding of how the MIT 2 is applied.

The goal of an observational study is to collect data about how a particular task is accomplished. We performed an observational study with the purpose of eliciting the process employed by software engineers when applying the current version of the MIT 2 technique. Our goal was to deeply understand the MIT 2 process, so we did not compare the MIT 2 with any other technique. The observational study should answer the following question: “Which steps the software engineers adopted in the MIT 2’s application?”.

Observational techniques can be employed to understand current work practices [17]. In this study, we gathered two types of data: observational and inquisitive data. The observational data were collected during the inspection process. To gather the observational data, we used the “Cooperative Evaluation” method [18]. In this method, the software engineer describes what (s)he is doing and the observer is free to ask questions/explanations about the software engineer’s decisions or actions [2]. Inquisitive data were gathered after finishing the inspection using interviews.

A. Planning

Quantitative data were measured in order to compare the quantitative results of this study with other studies conducted with the MIT 2 technique. The quantitative investigation points were the efficiency and effectiveness indicators of the technique. Efficiency and effectiveness were calculated for each subject as: (a) the ratio between the number of defects found and the time spent to find them; and (b) the ratio between

the number of detected defects and the total number of existing (known) defects, respectively.

The mockups used in this study are part of the SION System. The SION is a website that provides information about the advertisement and support of activities regarding Science, Technology and Innovation. The mockups that were evaluated in this study are: mockup of course registration (see Figure 1), mockup of course listing (where one can select a course to delete it or edit it), and some messages that the system displays after saving data. The mockups had real usability problems that would influence the use of the designed system.

The interviews and observations took place in a Formation Center from a large IT Company, where the focus is in innovation and software development. The center supports several software engineers in real projects, mainly in the development of Web systems and Mobile applications, adopting agile methodologies such as Scrum and XP. In order to meet ethical needs, we created a free consent form to inform about research procedures and confidentiality. Fifteen software engineers signed the consent form. All participants received one-hour training on mockups and usability principals. Examples were shown on how to use the MIT 2 technique.

The qualitative investigation points were Application Process and Intention to Use of Technique. Theses investigation points were collected during the study and were analyzed together with the data obtained from the interviews. For the interviews, a semi-structured questionnaire was used with open questions (see Table II)

TABLE II. INVESTIGATION POINTS RELATED TO THE QUESTIONS THAT WERE USED ON THE QUESTIONNAIRE.

Investigation Points	Questions
Application Process	How did you apply the technique regarding its reading order and looking for problems? Why do you think this is the best way to apply the technique?
	How did you apply the technique with respect to the order of using the heuristics? Why do you think this is the best way to apply the technique?
	How would you apply the technique if you were to carry out a new usability evaluation?
Intention to Use the Technique	Would you use this technique in a software development project on your work environment? How?

B. Execution

Each software engineer applied the MIT 2 technique, evaluating the mockup and identifying usability problems. When a software engineer found a usability problem, (s)he described the problem in a worksheet. After this, a researcher interviewed the software engineers and they provided their impressions regarding the MIT 2 technique. The observer provided forms containing some notes. It is important to notice that the observer could question the software engineer’s actions at any time, but (s)he was not allowed to help the software engineer in the discovery activity.

One of the researchers acted as the inspection’s moderator. The moderator was responsible for conducting the study. After the individual inspection by each software engineer, the moderator checked all discrepancies’ worksheets for incorrect

information and gathered the discrepancies. A discrepancy is an issue reported by the software engineer that could be a real defect or a false positive. During this activity, the moderator highlighted duplicated discrepancies.

After this, the discrimination meeting was executed by the moderator and two others researchers (not involved with the study). The purpose of this meeting was to analyze all discrepancies identified by each software engineer. The researchers verified if the discrepancy was a real defect or a false positive. It is worth mentioning that the researchers had high usability knowledge and prior experience in usability evaluations. The quantitative results of the discrimination meeting are presented in Section 5.

Finally, we transcribed the interviews to forms. The interviews allowed this research to gather information in order to understand how software engineers employed the MIT 2. The data analysis of these interviews is presented in Section 6.

V. QUANTITATIVE DATA ANALYSIS

After the discrimination activity, we counted the number of discrepancies, false-positives and defects, the time spent during the inspection, the efficiency and effectiveness per software engineer (see Table III)

TABLE III. SUMMARY OF INSPECTION RESULTS PER SUBJECT

Participants	Discrepancies	False Positive	Defects	Time (Hour)	Efficiency (Defects/Hour)	Effectiveness
P01	9	0	9	0.38	23.48	25.00%
P02	6	0	6	0.58	10.29	16.67%
P03	4	1	3	0.28	10.59	8.33%
P04	8	1	7	0.30	23.33	19.44%
P05	7	0	7	0.45	15.56	19.44%
P06	9	3	6	0.40	15.00	16.67%
P07	9	1	8	0.38	20.87	22.22%
P08	11	3	8	0.43	18.46	22.22%
P09	13	8	5	0.47	10.71	13.89%
P10	13	1	12	0.40	30.00	33.33%
P11	10	4	6	0.62	9.73	16.67%
P12	12	4	8	0.57	14.12	22.22%
P13	12	5	7	0.48	14.48	19.44%
P14	16	6	10	1.02	9.84	27.78%
P15	19	3	16	0.50	32.00	44.44%
Average	10.53	2.67	7.87	0.48	-	21.85%

Overall, the inspection resulted in a set of 36 usability defects, including the 7 seeded ones. Software engineers who used MIT 2 managed to find between 3 and 16 defects spending about 0.28 and 1.02 hours. The effectiveness in this observation study was 21,85%. Comparing this measure with the effectiveness of the group of undergraduate students who used the MIT 2 technique in the first feasibility study (16%) and in the second feasibility study (15,87%), we can notice that this measure was higher in the observation study.

VI. QUALITATIVE DATA ANALYSIS

After the quantitative analysis, we carried out a specific analysis of the qualitative data that were obtained through the comments of software engineers in an interview. These comments provide information such as difficulties and questions during the use of the technique. These issues pointed

us what parts of the technique need improvements. The qualitative analysis was based on the procedures of Grounded Theory (GT) [19].

The qualitative data collected through the interviews were analyzed using a subset of the stages of the coding process suggested by Strauss and Corbin [19] for the GT method: the open coding (1st phase) and axial coding (2nd phase). When analyzing the qualitative data, we created codes (relevant concepts to understand the perception on the technique and its use process) related to the speeches of the participants - open coding (1st phase). After this, the codes were grouped according to their properties, forming concepts that represent categories and subcategories. Finally, these codes were related to each other - axial coding (2nd phase). The goal of the analysis in this study was to understand how software engineers perform the application process of MIT 2. We decided not to elect a core category, because the GT rule is the circularity between the collection and analysis stages until the theoretical saturation is reached [19]. Therefore, the selective coding was not performed (3rd phase of the GT method). The steps of the open and axial coding were enough to understand why some problem occurred and how the inspection process is.

A. Point of View regarding the Application Process of MIT 2

This subsection presents the analysis of how the technique was applied in this study. Through the interviews we identified that the software engineers employed the MIT 2 in three different ways: (i) first, the software engineer read the technique and then looked for problems in the mockup (see quotation from P08 below); or (ii) first, the software engineer looked for the problem in the mockup and then (s)he read the technique (see quotation from P03 below); or (iii) initially the software engineer viewed the mockup, then (s)he read the technique and after this, (s)he changed the way in which the technique was applied, looking for problems as soon as an item was read (see quotation from P10 below).

“I read item by item and tried to find the problems in each mockup” (Participant 8).

“First, I gave a quick look to the mockups and I saw some problems that I knew. Then, I began to read the technique from the beginning” (Participant 3).

“As there are several items in the technique, I looked at the mockup and found some issues, but if I had to look for these issues (...) one by one I think it would take longer. So I preferred to keep these wrong things that I found and (...) I was doing the inspection in the order I had to read each item and found or related to something that (...) I had identified” (Participant 10).

Some opinions were also collected regarding the application of the technique. Some participants noted that when applying the MIT 2, after knowing it, it is better to skip some items (see quotation from P11 below). In addition, other participants said that first seeing the mockup and then reading the MIT 2 technique is not the best way to start the inspection (see quotation from P05 below). However, one of the participants believes that first observing the mockup and then reading the MIT 2, allowed him to think that applying the technique was easy because he knew where some of the problems were (see quotation from P06 below).

“I was skipping; there were even some [items] for which I did not find problems” (Participant 11).

“Looking [at the mockup] first I don’t think it is the best way to start the inspection” (Participant 5).

“During this evaluation I realized that I began to learn what this technique meant and I could look at one element and already know what problem was associated (...). Then this strategy of looking at the mockup and looking at the list can be a starting point for you to memorize the heuristics, but in the future what you realize is that you end up abandoning it and you develop a skill” (Participant 6).

Additionally, this study aimed at obtaining information about how participants would apply the MIT 2 if given the opportunity. Some participants would read the verification item and would already search for problems (see quotation from P03 below). The reasons given by the participants for this way of applying the technique are: if not employed that way (i) the inspection can be more time consuming, (ii) the inspector can forget the problems, (iii) the inspectors does not remember the item is and (iv) because it eliminates primary errors. One of the participants said that first (s)he would view the heuristics and then (s)he would look for the problem and only after (s)he found the problem, (s)he would look for the verification item related to it (see quotation from P14 below). However, other participants said that they would first analyze the mockups and then they would relate the problems encountered with technique (see quotation from P11 below).

“I think reading [the MIT 2] and then looking for the problems in the mockups would be the best way to do it [the inspection]” (Participant 3).

“If I were to carry out a new evaluation, I would not waste time reading it [MIT 2]. If I carried out an evaluation a second time, as I already have prior knowledge, I would look (...) in which of these heuristics (...) [the problem] it is related. But (...) to indicate the verification item, only the second time, checking the item number.” (Participant 14).

“The most appropriate way is to look at the mockups, analyzing it and relating it to the technique” (Participant 11).

B. Opinion about the Intention to Use the MIT 2 in projects

This subsection presents the participants' opinions regarding the use of the MIT 2 technique in software development projects. Some of the opinions were: (s)he would use the MIT 2 early in the project (see quotation from P04 below); and to apply to MIT 2 on a project it will be necessary to explain its advantage, because although there are cost of training and spending time, later there will be gain with the improvements (see quotation from P06 below).

“I would like to try using it [MIT 2] early in the project so I don’t carry it [the evaluation] out in the end” (Participant 4).

“[About the use of the technique in projects] (...) first there should be an explanation, an understanding that it will generate an additional cost to your development process. This cost can be time or human

resources and these impacts on the development of the software. So this cost has to be very well explained to the managers indicating that although you have an initial cost, you have a benefit short after. What is difficult is to convince people that this is important and that organizations can actually recognize such importance and assume that deadlines can be postponed or budgets reduced because of these improvements” (Participant 6).

VII. DISCUSSION

Regarding the Application Process of the MIT 2 investigation point (Subsection VI.A), it can be noted that there were participants who preferred to review the mockup first and then read the technique. When they read a verification item of MIT 2 and remembered a possible usability problem they already observed in the mockup, they related the problem with the verification item. For them, by reading the technique, there was the advantage of already knowing where some of the problems were. These software engineers also stated that it may be a starting point for the use of the technique, because this way of applying the technique also helps memorizing the items. However, other participants said they first observing the mockup and then reading the MIT 2 is not the best way to start the inspection. These participants read a verification item and started searched for the problem. For them, the inspection becomes faster, the software engineer does not forget to point out identified problems and where the verification item related to the problem are.

During the analysis of the study, the researchers noted that the technique helps identifying usability problems in the two ways of applying the technique. In addition, after using the technique for the first time, the software engineers gain prior knowledge of it, and can skip the reading of some heuristics. This way, there is no need to stipulate a prescribed order in the application process of the MIT 2. The software engineers tend to adjust the application of the technique to their own way of use. This allows software engineers to feel more comfortable using the technique according their convenience.

It can be noted that the second investigation point (Subsection VI.B) presented some opinions from the participants related to the use of the MIT 2 in projects. One of the participants indicated that in order to apply the MIT 2 in a project, it would be necessary to explain its advantage, because even though there will be costs in training and time, the software company will have gains with the improvements. For some software engineers, the usability evaluation in the early stages may allow advantages such as: less rework rate and lower costs. This is because the usability problems are identified earlier and repairs are carried out before the coding of the application. Fixing problems earlier is cheaper than correcting problems of something that has already been developed. Through this study with industry practitioners, software companies have evidence of the benefits and opinions of practitioners about early usability evaluation.

VIII. THREATS TO VALIDITY

As in all studies, there are threats that may affect the validity of the results [20]. In this section, we discuss the main

threats to validity of this study. Two main threats were considered that represent a risk for an inappropriate interpretation of the results: (1) training effects and (2) influence of the moderator. There may have been an effect of the training if the training regarding the MIT 2 was different in quality for each software engineer. We controlled the training effects by preparing a single training for all software engineers. Finally, to reduce the second threat, at the discrimination meeting, a team of experts made the analysis of the identified discrepancies, judging if they were usability defects or not, without interference from the moderator.

Three threats were considered regarding the generalization of our findings: (1) the validity of the evaluated artifact as a representative artifact; (2) the researcher inserted some defects in the mockups; and (3) participants with need for training. With regard to issue 1, the inspected mockups are part of the project for a real system. However, it is not possible to say that the mockup used represents all kinds of mockups. With regard to issue 2, all participants found every inserted usability problems. Furthermore, the number of defects found by the participants was much greater than the number of defects inserted by the moderator. With regard to issue 3, the ideal would be that there was no need for training. However, the short training time allows the technique to be used by software engineers with low experience in usability evaluations.

The main threats that may affect the ability to obtain correct conclusions in this study are the size and homogeneity of the sample. These are known problems in Software Engineering studies. Therefore, there are limitations in our results, which are considered indicators and not conclusive.

IX. CONCLUSIONS

This paper described an observational study aimed at eliciting the sequence of activities that is employed by software engineers when applying the MIT 2 technique. Both the qualitative and quantitative results of this study provided us with important feedback to improve the MIT 2 technique.

The qualitative analysis was based on the following investigation points: (1) the application process of MIT 2; and (2) the intention to use MIT 2 in development projects. The qualitative analysis showed that the 2 identified ways of applying MIT 2 in the study proved effective in detecting problems. Through these results, we also noticed that it is not necessary to define a predefined order of applying the MIT 2.

The quantitative analysis showed that the calculated effectiveness in this observational study (21.85%) was higher than the effectiveness measured in the feasibility studies, showing that the improvements made in the MIT 2 previously allowed it to support on the identification of more usability problems. However, other factors may have influenced this outcome, such as: (1) the knowledge increase regarding usability evaluations, (2) and the participants from the observational study were software engineers.

ACKNOWLEDGMENTS

This research is partially funded by the National Science Foundation (grant 1242257, Pan American Software Quality Institute). We would like to acknowledge the financial support granted by CAPES and FAPEAM through processes numbers:

062.00146/2012; 062.00600/2014; 062.00578 /2014; 01135/2011; and PAPE 004/2015. We also thank the participants in the observational study and researchers from USES Group, especially Ana Carolina Oran.

REFERENCES

- [1] A. De Angeli, M. Matera, M. Costabile, F. Garzotto, P. Paolini. "On the Advantages of a Systematic Inspection for Evaluating Hypermedia Usability". In *International Journal of Human-Computer Interaction*, v. 15(3), pp. 315-335, 2003.
- [2] F. Molina, A. Toval. "Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems". In *Journal Advances in Engineering Software*, v.10 (12), pp. 1306-1317, 2009.
- [3] K. Hornbæk, R. Høegh, M. Pedersen, J. Stage. "Use Case Evaluation (UCE): A Method for Early Usability Evaluation in Software Development". In *International Conference on Human-Computer Interaction*, pp. 578–591, 2007.
- [4] N. Juristo, A. Moreno, M. Sánchez, M. Baranauskas. "A Glass Box Design: Making the Impact of Usability on Software Development Visible". In *Conference on Human-Computer Interaction*, v. 4663, pp. 541-554, 2007.
- [5] G. Donahue. "Usability and the bottom line". In *Journal IEEE Software*, v.18 (1), pp. 31-37, 2001.
- [6] X. Ferré, N. Juristo, H. Windl, L. Constantine. "Usability Basics for Software Developers". In *Journal of IEEE Software*, v.18(1), pp. 22-29, 2001.
- [7] A. Seffah, E. Metzker. "The obstacles and myths of usability and software engineering". In *Communications of the ACM - The Blogosphere*, v. 47(12), pp. 71–76, 2004.
- [8] A. Fernandez, E. Insfran, S. Abrahão. "Usability evaluation methods for the web: A systematic mapping study". In *Information and Software Technology*, v.53 (8), pp. 789-817, 2011.
- [9] N. Valentim, T. Conte "Technical Report: Version 3 of MIT 2". Report Number 004. Available at: <http://uses.icomp.ufam.edu.br/attachments/article/42/RT-USES-2015-0004.pdf>, 2015.
- [10] S. Propp, G. Buchholz, P. Forbrig. "Integration of usability evaluation and model-based software development". In *Journal Advances in Engineering Software*, v. 40 (12), pp. 1223 – 1230, 2009.
- [11] J. Nielsen. "Heuristic evaluation". In *Usability Inspection Methods* (Eds. Nielsen and Mack), John Wiley & Sons, New York, 1994.
- [12] G. Travassos, F. Shull, J. Carver, V. Basili. "Reading Techniques for OO Design Inspections". University of Maryland, pp. 1 – 56, 2012.
- [13] E. Luna, J. Panach, J. Grigera, G. Rossi, O. Pastor. "Incorporating usability requirements in a test/model-driven web engineering approach". In *Journal of Web Engineering*, v. 9 (2), pp. 132-156, 2010.
- [14] F. Shull, J. Carver, G. Travassos. "An empirical methodology for introducing software processes". In *ACM SIGSOFT Software Engineering Notes*, v. 26, n. 5, pp. 288-296, 2001.
- [15] N. Valentim, K. Oliveira, T. Conte. "Defining an Approach for Usability Inspection in Design Models through Experimentation" (in Portuguese). In *Symposium on Human Factors in Computing Systems*, pp. 165-174, 2012.
- [16] N. Valentim, T. Conte. "Improving a Usability Inspection Technique based on Quantitative and Qualitative Analysis" (in Portuguese). In *Brazilian Symposium on Software Engineering*, pp. 171-180, 2014.
- [17] M. Maguire "Methods to support human-centred design". In *International Journal of Human Computer Studies*, v.55(4), pp. 587-634, 2001.
- [18] M. Müller, J. Haslwanter, T. Dayton. "Participatory Practices in the Software Lifecycle". In *Handbook of Human-Computer Interaction*, 2nd edition, Elsevier, pp. 255-297, 1997.
- [19] A. Strauss, J. Corbin. "Basics of Qualitative Research. Techniques and Procedures for Developing Grounded Theory". Sage Pub, 400 pg., 2008.
- [20] C. Wöhlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, A. Wessl. "Experimentation in software engineering: an introduction". Kluwer Academic Publishers, 2000.