A Controlled Experiment with Usability Inspection Techniques Applied to Use Case Specifications: Comparing the MIT 1 and the UCE Techniques

Natasha M. Costa Valentim, Jacilane Rabelo, Ana Carolina Oran, Tayana Conte
USES Research Group - Instituto de Computação
Universidade Federal do Amazonas
Manaus, Brazil
{natashavalentim, jaci.rabelo, ana.oran, tayana}@icomp.ufam.edu.br

Sabrina Marczak
MUNDDOS Research Group - Computer Science School
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre, Brazil
sabrina.marczak@pucrs.br

***Abstract*—A Use Case Model is composed of use cases that describe software functionalities through Use Case Specifications. The evaluation of the specifications that compose such a model can allow for an early identification of usability defects. We previously proposed MIT 1—Model Inspection Technique for Usability Evaluation that aims to support the identification of usability defects through the evaluation of use cases specifications. In this paper, we present the evaluation of this technique through a controlled experiment that measured its efficiency, effectiveness, perceived ease of use, and perceived usefulness when compared to the Use Case Evaluation (UCE) method. Our quantitative findings indicate that MIT 1 allows users to find more usability defects in less time than UCE. However, UCE was considered easiest to use and more useful than MIT 1, highlighting improvement needs for MIT 1.**

***Index Terms*—controlled experiment, use case model, use case specification, early usability, inspection, empirical study.**

I. Introduction

Models are a simplification of reality that helps us to comprehend and to analyze complex software systems [1]. One of the models built during the early stages of the development process is a Use Case Model, which is composed of a set of use cases that describe requirements functionalities through user case specifications. Da Cruz [2] argues that a use case model of a software system may be used as a model of the required system functionalities and the required constraints on the interaction between the user, identified as an actor, and the system itself. The specification of a use case is typically made through a textual description [2]. Use case specifications are an important resource to guide developers designing the system interactions with users as well as developing the system itself. This kind of specification has been suggested as a valuable means to integrate usability engineering directly into the software development process [3]. The evaluation of these specifications can allow for an early identification of usability defects that would otherwise be identified later on in the development life cycle. An early identification of the referred defects can promote an early discussion and solution of these defects, avoiding generating source code that will likely be wasted and will need to be reworked [4].

Early Usability, as it is called, can help reduce the number of usability-related defects detected in software development projects and can provide benefits such as higher user satisfaction [5, 6]. In its systematic mapping, Fernandez *et al.* [7] found that when usability defects 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.

In our systematic mapping of literature [8], we identified a method that supports the identification of usability defects through the inspection of use case specifications. The method named Use Case Evaluation (UCE) was proposed by Hornbæk *et al* [9] based on the Heuristic Evaluation method [10] and aims to facilitate the identification of usability defects at the point in the development process where the first key use cases are described in the development life cycle. UCE has a list of guidelines to assist the inspection.

However, when studying the UCE method in more details, we perceived that its guidelines are too general since they only offer recommendations in order to improve the usability through the specification of use cases. Such 'generality' may impose novice usability inspectors difficulties using the guidelines. Given this limitation, we decided to propose a usability inspection technique to help novice inspectors to more easily conduct usability inspections. Our proposed technique, named MIT 1 [11], is part of a set of techniques called Model Inspection Techniques for Usability Evaluation (MIT), including two complementary techniques: MIT 2—for usability inspection in mockups [12] and MIT 3—for usability inspection in activity diagrams [13]. MIT 1 provides a series of steps, called verification items, which can be used by novice inspectors to analyze use case specifications and identify usability defects aiming to increase the effectiveness and efficiency of inspections.

In order to analyze the performance of the MIT 1 technique compared to the UCE method and to identify which of the two techniques assist inspectors with little experience in usability to

find more usability defects—the one with guidelines (UCE) or the one with verification items (MIT 1), we conducted a controlled experiment aiming to empirically validate the Effectiveness, Efficiency, Perceived Ease of Use, and Perceived Usefulness of MIT 1 against UCE. This controlled experiment was conducted with undergraduate and graduate students from the Pontifical Catholic University of Rio Grande do Sul (PUCRS) in Brazil. Our preliminary findings indicate that MIT 1 exceeded UCE in terms of effectiveness and efficiency, meaning that MIT 1 allows users to find more usability issues in less time than UCE. However, UCE was considered easiest to use and more useful than MIT 1 by our study subjects. These findings suggest that MIT 1 is more appropriate when one wants to point out usability defects, however, that there is still the need for MIT 1 to be improved to make it easier for one to use it.

The remainder of this paper is structured as follows. Section 2 discusses the concept of Early Usability and briefly introduces the Heuristic Evaluation and UCE inspection methods. Section 3 provides an overview of our previously proposed MIT 1 inspection technique. Section 4 describes the planning and the execution of the controlled experiment. Section 5 presents the quantitative results of the experiment while Section 6 presents the perception of our subjects regarding MIT 1. Section 7 discusses some of the identified defects in the experiment. Section 8 describes possible threats to validity and Section 9 concludes the paper with our considerations about the findings and with a list of future work aiming to improve MIT 1.

II. Early Usability

Usability provides benefits such as the improvement of user productivity, the reduction with training, documentation costs [14], and income increase of software industry [15]. Therefore, a large number of researchers have investigated ways to address usability in early phases of software development [9, 16]. The aforementioned benefits of usability have motivated organizations to consider usability as a relevant factor in their software products [17].

However, when we include usability into the development process, we can observe the following: (a) usability activities are usually separated from the software development process [18], (b) development processes do not take advantage of the intermediate artifacts (e.g., navigational models and abstract user interface models) that are produced during early stages (i.e., requirements and design stages) [7]; and (c) notations and tools in which usability is represented and defined are different from those employed in software development processes [18].

Usability inspection methods that can be applied to models generated in early stages of the software development are fundamental for enabling better productivity and better user satisfaction in the final interfaces of software systems [14]. An overview of two usability inspection methods that can be applied to models of early stages is presented in the following sections.

A. Heuristic Evaluation Method (HEM)

The Heuristic Evaluation Method proposed by Nielsen [10] is an inspection method that is widely recognized and used by industry and by academia [14]. Nielsen defined his method based on extensive empirical evidence collected over the years. HEM covers a broader range of usability aspects and has often been used for comparison with others inspection methods [4]. Furthermore, HEM aims at finding usability defects by using a compliance analysis of the system using heuristics or quality standards [4].

Nielsen proposed ten heuristics that are intended to cover the best practices in the design of any user interface: (1) Visibility of system status; (2) Match between system and the real world; (3) User control and freedom; (4) Consistency and standards; (5) Error prevention; (6) Recognition rather than recall; (7) Flexibility and efficiency of use; (8) Aesthetic and minimalist design; (9) Help users recognize, diagnose, and recover from errors; (10) Help and documentation.

B. Use Case Evaluation (UCE)

Hornbæk *et al.* [9] propose the UCE method for the evaluation of usability in software systems based on the inspection of use cases. The method employs Nielsen's [10] heuristics. The UCE method consists of three activities: (1) Inspection of Use Cases, it seeks to identify usability defects that the evaluator is convinced one or more prospective users will experience; (2) Assessment of Use Cases, it seeks to assess the quality of the use cases; and (3) Documentation of Evaluation, where the results are compiled into a coherent evaluation of the product.

UCE has 11 heuristics, named as follows: (1) Visibility of system status; (2) Match between system and the real world; (3) User control and freedom; (4) Consistency and standards; (5) Error prevention; (6) Recognition rather than recall; (7) Flexibility and efficiency of use; (8) Help users recognize, diagnose, and recover from errors; (9) Avoid hard mental operations and lower workload; (10) Avoid forcing the user to premature commitment; and (11) Provide functions that are of utility to the user. Table 1 presents some UCE heuristics.

TABLE 1. EXAMPLE OF UCE HEURISTICS – HEURISTICS 1 TO 4 [9]

Heuristic 1. Visibility of system status
The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
Heuristic 2. Match between system and the real world
The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system oriented terms. Follow real-world conventions and make information appear in a natural order.
Heuristic 3. User control and freedom
Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
Heuristic 4. Consistency and standards
Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

III. Model Inspection Technique 1 (MIT 1)

MIT 1 aims to increase the effectiveness of inspections, providing guidelines that can be used by inspectors to analyze the use cases specification and identify usability defects [11]. Therefore, MIT 1 has verification items that serve as a guide to interpret Nielsen's heuristics [10] when applied to use case specification. These steps are: (1) to evaluate the use case, the inspector must check if the use case specification meets each of the usability verification items and (2) to identify usability defects, the inspector must point in the use case specification which part did not meet the usability verification items. Table 2 shows examples of the verification items.

MIT 1 is divided into high and low detailed level, respectively for use cases with high and low level of details. The MIT 1 technique – High Detailed Level is used for inspecting use cases that present information such as error messages, informational texts, warnings, name of screen, name of fields, among others. The MIT 1 technique – Low Detailed, on the other hand, is used for inspecting use cases that do not present such information. The advantage of having such division is that inspectors do not have to waste time reading verification items that will not help them finding defects for a particular type of use case given the referred details are not present. The full version of MIT 1 is available online in a technical report [19]. We presented examples of usability defects identified with MIT 1 in Section 7.

MIT 1 was evaluated in comparison to HEM [10] in previous studies [11]. We chose HEM because: (a) MIT 1 is based on HEM and (b) HEM is an inspection method widely used in industry and academia [14]. In our previous studies, we verified that MIT 1 showed slightly better efficiency than HEM However, no statistically significant difference was found. Regarding effectiveness, the group that used MIT 1 had a significantly higher performance than the one that used HEM.

Table 2. Example of MIT 1' verification items [19]

MIT-1AA. Visibility of system status	
Verification Item 1AA1	Verify if there is some text in the Main, Alternative and Exception Flows which informs where in the system the user is.
Verification Item 1AA2	Verify if there is some text in the Main, Alternative and Exception Flows which informs the user what was done after data persistence. For example, when changing or deleting something, a text message is displayed.
MIT-1AB. Heuristic Match between system and the real world	
Verification Item 1AB1	Verify if the names of fields, screens, buttons, links, error messages and informational texts in the Main, Alternative, Exception Flows and Business Rules have familiar concepts to users, i.e., follows the conventions of the real world.
Verification Item 1AB2	Verify if the options, screens or fields reported by the system in the Main, Alternative and Exception Flows are presented in a natural and logical order according to the concepts of the defect domain.
MIT-1AC. Heuristic User control and freedom	
Verification Item 1AC1	Verify if the user, through Alternative and Exception Flows, can undo or redo an action involving persistent data in the system. For example, to check if one can delete or change entered data.

In order to check MIT 1 performance compared to a technique that has the same purpose (to evaluate the usability through use case specifications), we conducted a controlled experiment comparing the MIT 1 technique with the UCE method. We described this experiment below.

IV. Controlled Experiment

This experiment aimed at empirically validating the Effectiveness, Efficiency, Perceived Ease of Use, and Perceived Usefulness of MIT 1 when compared to UCE.

This controlled experiment followed the inspection process suggested by Sauer *et al*. [20]. This process was divided into four activities that are presented in Fig. 1. The activities and roles that make up the usability inspection process will be described in the following sections.

A. Planning

Planning of inspection, the first activity of the inspection process based on Sauer *et al.* [20], is the activity in which the definition of the experiment scope, material preparation, selection of subjects, training of subjects in the techniques, and assignment of tasks to each subject is made. The inspection leader (or moderator) carries out this activity. In our experiment, a person who has knowledge and experience in usability evaluations carried out this activity.

1) *Hypotheses*: The experiment was planned and conducted in order to test the following hypotheses (null and alternative, respectively):

- H01: There is no difference between the MIT 1 and UCE techniques regarding the efficiency indicator.
- HA1: There is a difference in the efficiency indicator when comparing the MIT 1 and the UCE techniques.
- H02: There is no difference between the MIT 1 and UCE techniques regarding the effectiveness indicator.
- HA2: There is a difference in the effectiveness indicator when comparing the MIT 1 and the UCE techniques.

2) Context: We carried out the experiment with one of the use cases of an online system for showing indicators of research and development in Brazil (see an extract in Fig. 2). The experiment was conducted with 4th year undergraduate students of the Computer Science course and graduate students at the Computer Science Master and Doctorate degrees at Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), one of the largest private university in Brazil.

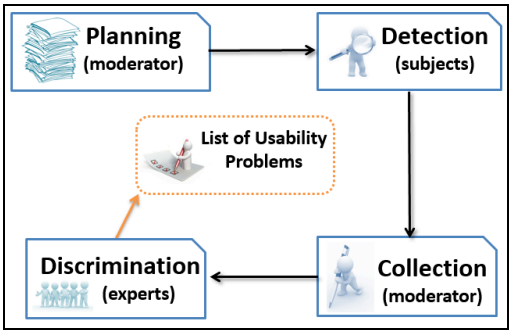

Fig. 1. Inspection process baseed on Sauer *et al.* [20]

Alternative Flow - Edit a course

1. The system displays the screen "Course Registration/Edition - Training Center".
2. The employee clicks on Edit button. This button is beside to the desired course.
3. The system shows the screen "Course Registration - Training Center".
4. The employee changes the course name in the Course field.
5. The employee selects an option from the Segment field.
6. The system updates responsible in the Responsible field.
7. The employee clicks on Go button.
8. The system displays message: "The course was changed."

Fig. 2. Part of use case specification used in the experiment

3) *Variables Selection*: The independent variables were the usability evaluation techniques (MIT 1 and UCE) and the dependent variables were the efficiency and effectiveness indicators of the techniques. Efficiency and effectiveness were calculated for each subject as: (a) the ratio between the number of defects detected and the time spent in the inspection process; and (b) the ratio between the number of detected defects and the total number of existing (known) defects, respectively.

4) *Selection of Subjects:* We did not establish any aim for the number of subjects for the experiment, but we tried to use the maximum number of students available in order to detect a representative number of usability defects. Forty-eight (out of 54) students consented to participate in the study. They signed a consent form and filled out a characterization form that measured their expertise with usability evaluation (UE) and software development (SD). The characterization form was employed to categorize the subjects as having: none, low, medium or high experience regarding usability evaluation and software development. We considered:

- Highly experienced (H): subjects who had participated in more than 5 usability projects/evaluations in industry;
- Medium experienced (M): subjects who had participated from 1 to 4 usability projects/evaluations in industry;
- Low experienced (L): subjects who participated in at least one usability project/evaluation in the classroom and;
- With no experience (N): subjects who had no prior knowledge about usability or who had some usability concepts acquired through lectures/speeches but no practical experience.

Analogously, the subjects' expertise in software development was classified following the same standards. Second (UE) and third (SD) columns of Table 3, presented in Section 5, show each subject's categorization respectively.

With regard to undergraduate students participation, all were given four points in their midterm (equivalent to 0.5 points of their course final grade) regardless of their performance given that they are all part-time students and had to come to University for 3 extra hours. With regard to graduate students participation, the participation was voluntary and the experiment took place during class hours.

5) *Experimental Design*: Subjects were divided into two groups, which would inspect the same use case: the MIT 1's group and the UCE's group. The subjects were assigned to each technique using completely randomized design. Each group was composed of 24 subjects. However, 3 subjects allocated to the UCE group did not attend the experiment.

6) *Instrumentation*: Several artifacts were defined to support the experiment: characterization and consent forms, specification of the UCE and MIT 1 techniques, instructions for the experiment, a worksheet for the annotation of the identified discrepancies and a post-inspection questionnaire. In addition, we used a use case that is part of the specification of a real system from a Training Center that manages courses. See part of use case specification in Fig 2. The authors of this paper validated all artifacts.

7) *Preparation*: All subjects received a one-hour training on usability evaluation. The training material included a set of slides containing an introduction to usability in order to present the basic concepts. Additionally, for each group, we made a 15-min presentation about the technique that the group would apply. Similar examples were shown on how to use both techniques (MIT 1 and UCE).

B. Detection

The second activity of the inspection process based on Sauer *et al.* [20] is the detection of defects, in which each inspector seeks usability defects in models, individually.

At the beginning of the experiment, a researcher (2nd author) acted as moderator, being responsible for passing the information about the expected evaluation to the inspectors, the students. Then, we divided the subjects into groups for each technique and each group went to work in a different classroom supervised by the 2nd and last authors. Each subject received the artifacts described in Section 4.6, Instrumentation. During the inspection, each subject filled out a worksheet with the identified candidate defects. All subjects returned the worksheet containing the identified defects and the indication of the total time spent in the inspection. They also filled out the follow-up questionnaire. Each inspector carried out the defect detection activity individually. During the detection activity, inspectors did not receive any assistance from the researchers involved in the experiment.

C. Collection

After the detection, the moderator performed the collection, third activity of the inspection process, where the lists of individual discrepancies were integrated into a single list, removing the reference to the inspector who found the discrepancy and the technique she had applied. This activity was conducted by the first author.

D. Discrimination

The fourth and last activity of the inspection process is the discrimination. In this activity, a team formed by three usability

experts reviewed such list. This team decided which of the discrepancies were unique and which were duplicated. Duplicated discrepancies were equivalent discrepancies pointed out by more than one inspector. Also, the team decided which discrepancies were real defects or false positives defined as detected usability defects considered as 'not real' defects.

V. Quantitative Results

Table 3 shows the overall results of the usability evaluation in use cases per subject. The label 'S' and a number identify each subject, e.g. S01 identifies subject 01.

TABLE 3. SUMMARY OF INSPECTION RESULT PER SUBJECT

Sub.	UE	SD	DC	FP	DF	Time (hour)	Defects/ Hour	Tech.
S01	L	N	12	0	12	0.78	15.32	MIT 1
S02	L	L	14	0	14	0.92	15.27	
S03	L	H	9	0	9	1.17	7.71	
S04	L	L	14	0	14	0.78	17.87	
S05	L	L	22	2	20	0.75	26.67	
S06	M	M	19	4	15	0.70	21.43	
S07	N	H	24	17	7	0.88	7.92	
S08	N	N	24	9	15	1.33	11.25	
S09	L	M	10	2	8	0.50	16.00	
S10	L	H	17	3	14	0.83	16.80	
S11	L	L	25	9	16	0.50	32.00	
S12	L	M	21	1	20	0.52	38.71	
S13	N	L	12	1	11	0.63	17.37	
S14	L	M	12	0	12	0.67	18.00	
S15	N	M	10	0	10	0.67	15.00	
S16	L	M	8	0	8	0.58	13.71	
S17	L	H	8	0	8	0.32	25.26	
S18	N	M	19	1	18	0.58	30.86	
S19	N	H	7	3	4	0.67	6.00	
S20	L	L	15	0	15	0.75	20.00	
S21	L	M	11	0	11	0.50	22.00	
S22	L	M	21	0	21	0.75	28.00	
S23	L	M	16	0	16	0.73	21.82	
S24	N	M	14	1	13	0.57	22.94	
S25	M	M	9	3	6	0.67	9.00	UCE
S26	L	M	14	0	14	0.75	18.67	
S27	N	H	6	0	6	0.35	17.14	
S28	N	H	15	1	14	0.50	28.00	
S29	N	M	10	1	9	0.48	18.62	
S30	H	H	17	1	16	0.78	20.43	
S31	L	M	5	0	5	0.67	7.50	
S32	L	M	11	2	9	0.75	12.00	
S33	N	M	11	1	10	0.50	20.00	
S34	N	N	7	1	6	0.50	12.00	
S35	L	M	12	2	10	0.48	20.69	
S36	N	M	8	2	6	0.52	11.61	
S37	N	L	8	3	5	0.78	6.38	
S38	L	L	7	2	5	0.63	7.89	
S39	M	M	5	1	4	0.27	15.00	
S40	L	L	12	3	9	0.58	15.43	
S41	L	M	11	5	6	0.60	10.00	
S42	N	M	10	0	10	0.75	13.33	
S43	L	N	8	3	5	0.50	10.00	
S44	L	M	8	0	8	0.67	12.00	
S45	L	M	8	0	6	0.25	24.00	

Legend:
Sub. – subject; **UE -** Experience in Usability Evaluation; **SD** - Experience in Software Development; **H -** High**; M** - Medium; **L -** Low; **N -** None; **DC -** Number of Discrepancies; **FP -** Number of False Positives; **DF -** Number of Defects; **Tech. –** Technique.

We can see that inspectors who used MIT 1 managed to find between 4 and 21 defects spending about 0.32 and 1.33 hours. On the other hand, the inspectors that used UCE employed between 0.25 and 0.78 hours, however they found between 4 and 16 defects. It can be noted that MIT 1 helped to identify more defects than UCE. However, the inspectors took more time using MIT 1 as per the applied statistical tests as described below.

Overall, the inspections resulted a set of 113 usability defects (known), including the 11 seeded ones (defects inserted by the moderator). Table 4 presents the average effectiveness and efficiency. The effectiveness of MIT 1 in this experiment was 11.47%. Comparing this measure with the effectiveness of the group who used the UCE method (7.12%), we can notice that this measure was higher.

Moreover, it can be observed in Table 4 that UCE tends to provide a low degree of false positives, total of 31 false positives compared to 53 false positives of MIT 1. The low degree of false positives can be explained by the fact that UCE provides a more simple procedure (less content) to detect usability defects. However, it can be observed that MIT 1 supported to find more usability defects (311 defects including the duplicates) than the UCE method (169 defects including the duplicates). The high degree of defects found with the MIT 1 technique can be explained by the fact that MIT 1 guides more the subjects on identifying usability defects.

We present the summary of the quantitative results per indicator using a boxplot graph. The statistical analysis was carried out using the statistical tool SPSS V. 22, and $\alpha = 0.05$. Figure 3 shows the boxplot graph with the distribution of efficiency per technique.

From Fig. 3, it can be observed that the MIT 1's group had almost the same efficiency as the UCE's group, since MIT 1's group median is almost in the same level than UCE's group median. The number 12 in the Fig. 3 represents the subject who had the best performance in this indicator related to the MIT 1 group. We used the test Shapiro-Wilk to test the normality. This test is indicated for sample with size less than 50 [21]. We verified that Efficiency was normally distributed (p-value = 0.147). In order to determine whether the difference between the samples is significant, we applied the parametric *t*-test [22] for independent samples. When we compared the two samples using the *t*-test, we found significant differences between the two groups (p-value = 0.030). These results support the rejection of the null hypothesis H01 (p-value < 0.05), and the acceptance of its alternative hypothesis HA1, suggesting that MIT 1 was more efficient than UCE when used to inspect the specification of the use case in this experiment.

TABLE 4. EFFECTIVENESS AND EFFICIENCY PER TECHNIQUE

Technique	MIT 1	UCE
Total Defects	311	169
Average Defects	12.96	8.05
Total False Positives	53	31
Effectiveness	11.47%	7.12%
Average Time (hour)	17.08	11.98
Efficiency (defects/hour)	18.20	14.10

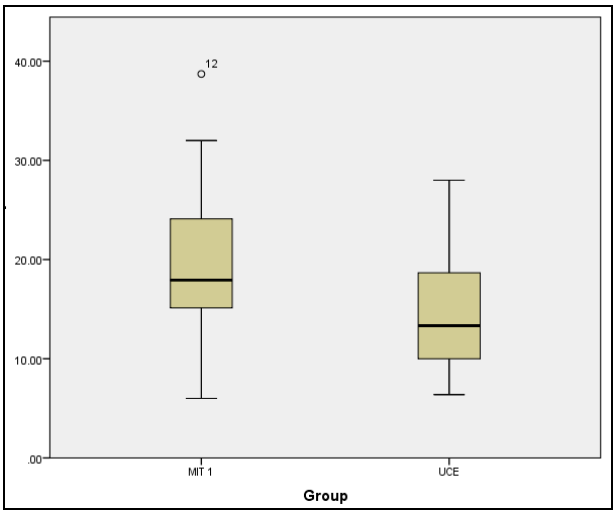

Fig. 3. Boxplots for efficiency

The same analysis was applied to determine whether there was a significant difference comparing the effectiveness indicator of the two techniques in detecting usability defects. The boxplots graph with the distribution of effectiveness per technique (see Fig. 4) suggests that the MIT 1's group was much more effective than UCE's group when inspecting the usability of the use case. Also, MIT 1's group median is much higher than UCE's group median, and all of the MIT 1's group boxplot is above UCE's group third boxplot quartile. The number 30 in the Fig. 4 represents the subject who had the best performance in this indicator related to the UCE group. We also used the test Shapiro-Wilk to test the normality and we verified that Effectiveness was not normally distributed (p-value = 0.034). In order to determine whether the difference between the samples is significant, we applied the Mann-Whitney test [23]. The p-value obtained in the Mann-Whitney test was 0.00. This result therefore supports the rejection of the null hypothesis H02 (p-value < 0.05), and the acceptance of its alternative hypothesis HA2, suggesting that the MIT 1 technique was more effective than UCE when used to inspect the specification of the use case in this experiment.

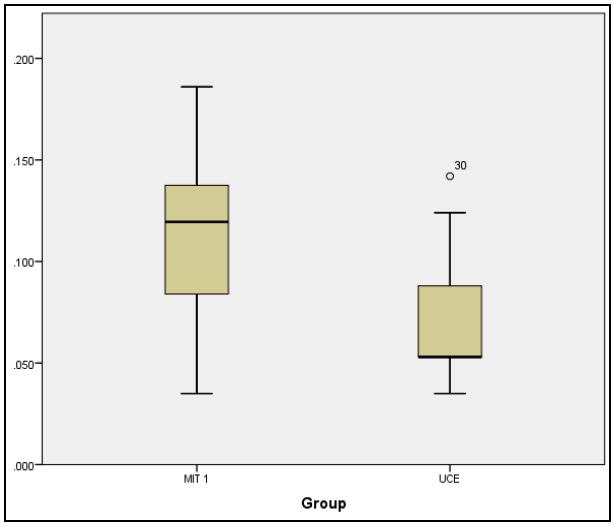

Fig. 4. Boxplots for effectiveness

A correlation analysis was performed with the variables effectiveness and efficiency to determine the relationship between them. The Spearman correlation coefficient between effectiveness and efficiency was 0.726, with p-value = 0.00, showing positive correlation. Therefore, subjects more effective were also more efficient.

VI. Analysis of User Perception

As indicated by Hornbæk [24], for assessing the quality of usability evaluation methods, not only the counting of usability defects detected should be considered but also the user satisfaction with the methods under evaluation.

Therefore, after the quantitative analysis, the post-inspection questionnaires about technique acceptance concerning MIT 1 and UCE were analyzed. Such questionnaires have been defined based on the indicators of Technology Acceptance Model – TAM [25]. The indicators used were: (i) perceived ease of use and (ii) perceived usefulness. The reason for focusing on these indicators is that these aspects are strongly correlated to user acceptance.

Subjects provided their answers in a six-point scale, based on the questionnaires applied by Lanubile *et al.* [26]. The possible answers are: totally agree, strongly agree, partially agree, partially disagree, strongly disagree, and totally disagree. This scale was considered appropriate to our goal because there is no middle value, i.e., it helps to avoid central tendency bias in ratings by forcing raters to judge the output as either adequate or not adequate [27, 28, 29].

A. Perceived Ease of Use

Figure 5 presents the perceptions of the subjects regarding the ease of use of the UCE and MIT 1. The X-axis of the graphs refers to the possible answers of the post-inspection questionnaire and the Y-axis refers to the number of subjects. S01, S02, S03, and so on, are codes to indicate the subjects presented in Table 3.

It can be noted that subject S13 totally disagreed in relation to the statement *"I managed to use the technique in the way that I wanted"*. Subject S13 was rated with no experience in usability evaluation (see Table 3). Moreover, the same subject totally disagreed in the statement *"I consider that the technique is easy to use"*. This can indicate that the MIT 1 technique is not suitable for people with little or no experience in usability and must be improved to become more ease to use.

The statement *"It's easy to remember how to use the technique to conduct a usability inspection"* obtained disagreements both the MIT 1 subjects (S1, S4, S13, S20) as UCE (S28, S33, S43), showing that it is not easy to remember both techniques. It is noteworthy that most of these subjects have low experience with usability; therefore, it is necessary that techniques minimize the subjects' memory load.

B. Perceived Usefulness

Figure 6 presents the subjects' perceptions regarding the usefulness of MIT 1 and UCE. We can verify that subject S02 totally disagreed in relation to the statements *"The technique allowed to detect defects faster"* and *"Using technique improved my performance in the inspection"*.

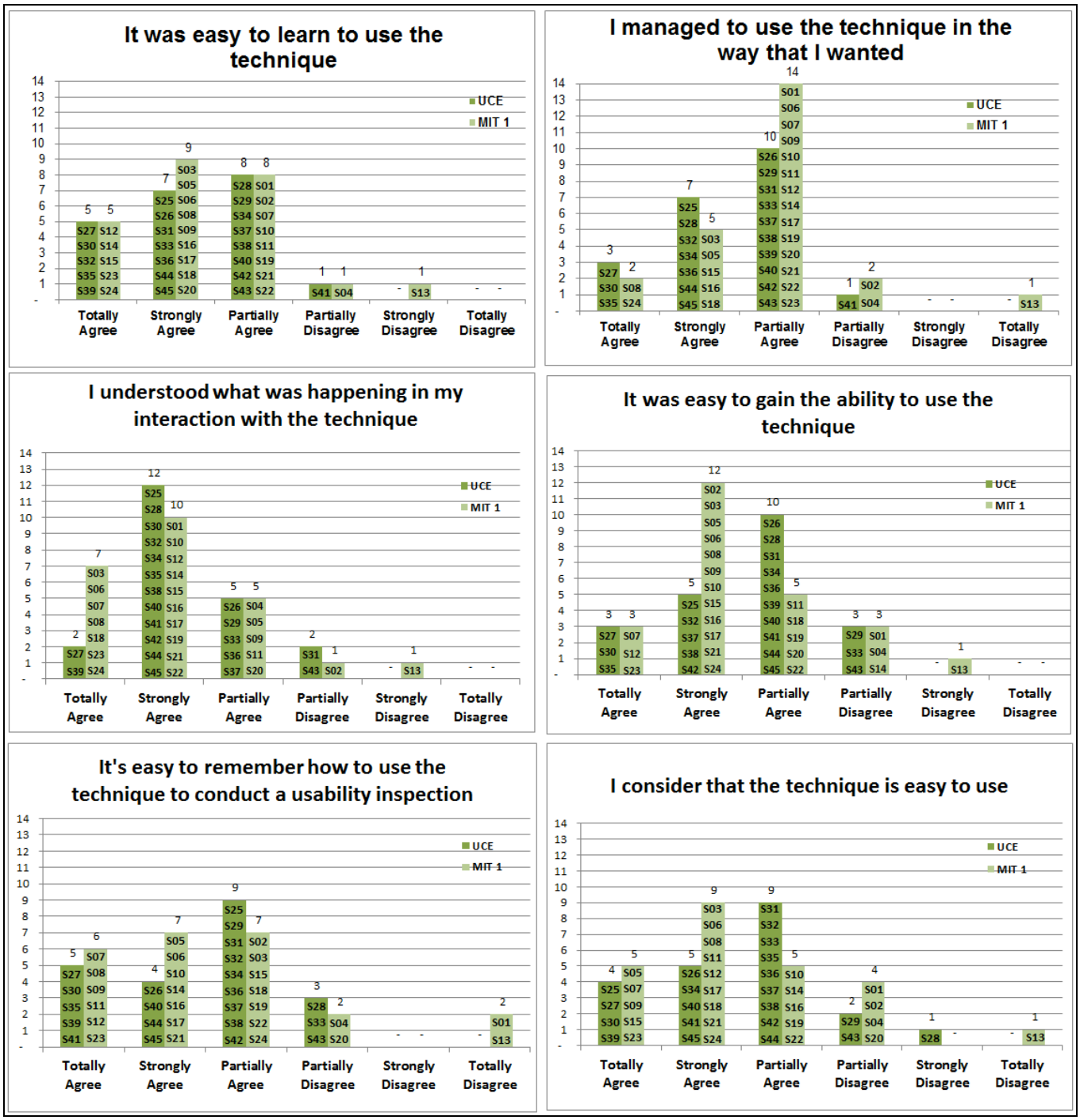

Fig. 5. Subjects' perception on ease of use of MIT 1 and UCE

Subject S02 had low experience in usability. It took about 0.92 hour to perform the inspection, finding around 14 usability defects. The performance of subject S02 as well as the time spent on inspection was fairly reasonable, but for the subject, MIT 1 should help to find defects faster.

VII. Discussion

We performed an analysis of the defects that were found by the MIT 1 group and that were not found by the UCE group and vice versa. Due to space limitations, this paper only presents some of defects found in the use case specification.

Subject S20 (MIT 1 group) reported the following defect *"The system does not allow user to remake any changes in data"*. This subject reported in the discrepancy worksheet that he identified this defect through verification item 1AC1 of the MIT 1 technique that requests the following *"Verify if the user, through Alternative Flow and Exception Flow, can undo or redo something involving persistent data in the system. For example, one can delete or change entered data"*.

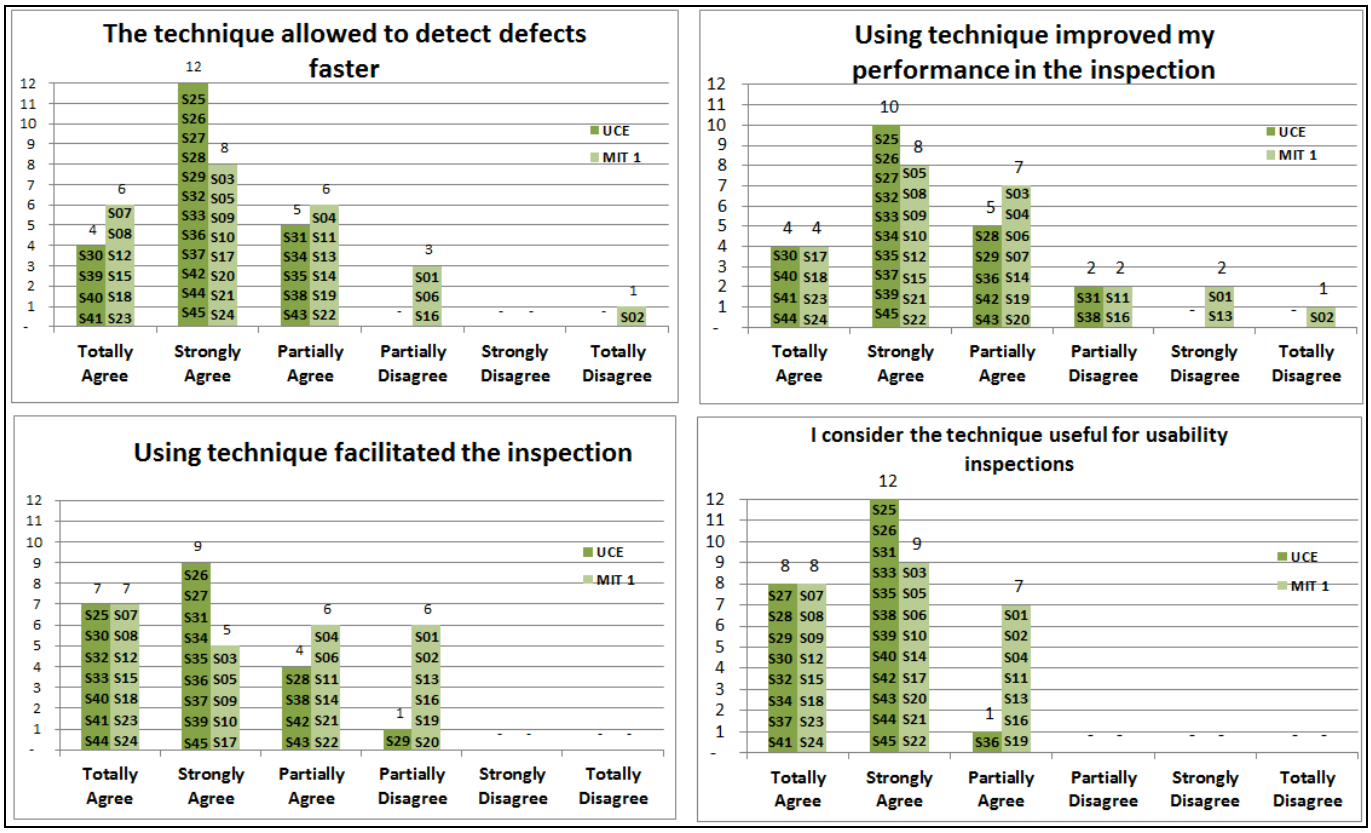

Fig. 6. Subjects' perception on usefulness of MIT 1 and UCE

This defect was not reported by any of the subjects of the UCE group. This could indicate that UCE needs to have more guidance related to identify defects related to the "control and freedom of the user" heuristic.

Another subject of the MIT 1 group (S04) wrote the following defect *"The screen "Course Registration/Editing – Training Center" brings two possibilities, giving a possible ambiguity to the user"*. This defect was identified by the verification item 1AD3 *"Verify if the names of fields, screens, buttons and links in the Main Flow, Alternative Flow, Exception Flow and Business Rules have a unique sense, without ambiguities"*. None of the subjects of the UCE group identified this defect. This may suggest that to have verification items (as MIT 1) guide more inspectors to find more usability defects. Note that the verification item 1AD3 presented above is related to heuristic Consistency and Standards. Therefore, the UCE method should guide more inspectors to identify usability defects related to this heuristic.

Subject S41, who used the method UCE, identified the following defect *"Absence of instructions to the user of the significance of options in Segment field"*. He identified this defect using heuristic Error prevention that guides the following *"Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Eliminate error-prone conditions or handle them gracefully"*. The lack of significance of the options in Segment field can lead the user to error. This type of defect was not reported by any of the MIT 1 subjects. This may indicate that it is necessary to include verification items that guide the user to identify this type of defect in use case specifications in the MIT 1 technique.

The analysis of the defects encountered with MIT 1 and not encountered with the UCE and vice versa in each flow of use case specification allows the identification of improvements in both techniques. However, our main concern

is identifying improvement opportunities for MIT 1.

VIII. Threats to Validity

As in all experiments, there are threats that could affect the validity of results. In this section we discuss the threats to the validity of our findings. We categorized them as per approach as Wohlin *et al*. [30] definitions: internal, external, conclusion, and construct threats.

A. Internal Validity

In our experiment, we considered four main threats that represent a risk for an improper interpretation of the results: (1) training effects, (2) experience classification, (3) time measurement, and (4) influence of the moderator. There might be a training effect if the training on the UCE technique was of lower quality than the training on the MIT 1 technique. We controlled training effects by preparing equivalent training courses with the same examples of discrepancies detection. Also, regarding subject experience classification, this was based on the subjects' self-classification. They were classified according to the number and type of previous experiences in

usability evaluation and software development. Considering time measurement, we asked the subjects to be as precise as possible, and the moderator also checked the time noted by each subject when she delivered her worksheet. Finally, to reduce the threat regarding the influence of the moderator on the results of the experiment, a team of experts did an analysis over the identified discrepancies. Such team judged whether the discrepancies were usability defects or not, without the interference from the moderator.

B. External Validity

Five issues were considered: (1) subjects were undergraduate and graduate students, (2) the experiment was conducted in an academic environment, (3) the validity of the evaluated model as a representative model, (4) some defects were seeded in the model, and (5) subjects required training. Regarding Issue 1, under certain conditions, there is no great difference between this type of students and professionals [31, 32], and they could therefore be considered as the next generation of professionals [33]. In addition, according to Carver *et al.* [32], students who do not have experience in industry may have similar skills as less experienced inspectors. Regarding Issue 2, the inspected artifact (use case) is a model that is part of the specification of a real system. However, it is not possible to state that the model used within the inspection represents all types of use cases (Issue 3). Regarding Issue 4, all seeded usability defects were found by both groups of subjects. Furthermore, the number of defects found by the inspectors in both groups was much larger than the number of defects seeded by the searcher. Finally, regarding Issue 5, it would be ideal if there was no training needed in order to apply the technique. However, the short time spent in training allows developers to use the technique without prior experience in usability evaluation.

C. Conclusion Validity

The main threats to the conclusion validity were the data collection and the homogeneity of the sample. With regard to the data collection, we applied the same procedure in each individual experiment in order to extract the data, and ensured that each measure was calculated by applying the same formula. With regard to homogeneity of the sample, the subjects are all students from the same institution. Due to this fact, there is a limitation in the results, which should be considered indicators and not conclusive ones.

D. Construct Validity

The construct validity may have been influenced by the measures that were applied in the quantitative analysis and in the user' perceptions. We intended to alleviate the first threat by evaluating the measures that are commonly employed in experiments in which usability inspection methods are involved. In particular, we employed the Effectiveness and Efficiency measures as suggested by Hartson *et al.* [34] for formative evaluations (i.e., usability evaluations during the development process). These measures have also been employed in similar experiments [6, 14].

IX. Conclusion and Future Work

The lack of usability evaluation methods with detailed instruction of use that can properly be integrated into early stages of development processes motivated us to propose, in a previous work, the Model Inspection Technique for Usability Evaluation (MIT 1) [11] as an inspection method which can be used to evaluate the usability through use case specifications.

In this paper, we report on the results of a controlled experiment that aimed at evaluating subjects' effectiveness, efficiency, perceived ease of use, and perceived satisfaction of use when using MIT 1 in comparison to a inspection method also based on heuristics, the Use Case Evaluation (UCE) method. We know that heuristics enable a person to discover or learn something for oneself [4]; therefore, there are no specific instructions on how to use them. MIT 1 although also based in heuristics, uses verification items to express them and instruct the inspector on how to proceed. Verification items are commonly used as a mechanism to express how an inspection technique should be applied. The comparison of inspection techniques using distinct formats is ordinary in literature [35].

The results of the quantitative analysis showed that MIT 1 was more effective than UCE in the detection of usability defects in use case specification. We verified that the UCE subjects completed the task faster; however, the MIT 1 subjects found more defects. By considering the results of the efficiency indicator as a whole, we conclude that MIT 1 was more efficient than UCE. In addition, a correlation analysis was performed, showing positive correlation. Therefore, subjects more effective were also more efficient.

The low ratio of false positives obtained by UCE group suggests that the use of technique with less content reduces the degree of subjectivity in the evaluation of use case specification. However, the high ratio of defects found with MIT 1 suggests that the use of technique with more guidance increases the number of usability defects identified.

With regard to the subjects' perceptions, the subjects were more satisfied when they applied UCE, and they also found it easier to use than MIT 1, showing that MIT 1 needs to be improved in this sense.

Some discussions were raised with regard to defects found by one technique (MIT 1) and not found by the other (UCE) and vice versa. This allows the identification of improvements that should be made in both techniques in order to obtain a more complete technique that assists in finding a more usability defects.

As future work, we intend to implement improvements in the MIT 1 technique and to perform more replications in order to minimize the influence of the threats to validity identified. In particular, these replications will consider: different experimental designs, new kinds of subjects such as practitioners from industry with different levels of experience in usability evaluations, and other use case specifications from different development processes.

Acknowledgment

We thank the undergraduate and graduate students from Pontifícia Universidade Católica do Rio Grande do Sul who

participated in the experiment. We would like to acknowledge the financial support granted by CAPES (Foundation for the Improvement of Highly Educated Personnel) and FAPEAM (Foundation for Research Support of the Amazonas State) through processes numbers: 062.00600/2014; 062.00578/2014; and 01135/2011.

REFERENCES

[1] V. Sien, "Teaching Object-Oriented Modelling using Concept Maps". In Journal Electronic Communications of the European Association of Software Science and Technology, 2010, v. 34, pp. 1-13.

[2] A. da Cruz. "Refining Use Cases Through Temporal Relations". In 9th International Conference on Software Paradigm Trends, 2014, v. 1, pp. 95-102.

[3] X. Ferré, N. Juristo, H. Windl, L. Constabile. "Usability Basics for Software Developers". In Journal of IEEE Software, 2001, v. 18 (1), pp. 22-29.

[4] A. Fernandez, S. Abrahão, E. Insfran. "Empirical validation of a usability inspection method for model-driven Web development". In Journal of Systems and Software, 2013, v.86 (1), pp. 161-186.

[5] 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, 2009, v.10 (12), pp. 1306-1317.

[6] A. Fernandez, S. Abrahão, E. Insfran, M. Matera. "Usability Inspection in Model-driven Web Development: Empirical Validation in WebML". In Proceedings of 16th International Conference on Model Driven Engineering Languages and Systems, 2013, pp. 740-756.

[7] A. Fernandez, E. Insfran, S. Abrahão. "Usability evaluation methods for the web: A systematic mapping study". In Information and Software Technology, 2011, v.53, pp. 789-817.

[8] W. Silva, N. M. C. Valentim, T. Conte. "Integrating the Usability into the Software Development Process: A Systematic Mapping Study". In Proceedings of the 17th International Conference on Enterprise Information Systems, 2015, pp. 105-113.

[9] K. Hornbæk, R. T. Høegh, M. B. Pedersen, J. Stage. "Use Case Evaluation (UCE): A Method for Early Usability Evaluation in Software Development". In Proceedings of the 11th International Conference on Human- Computer Interaction, Rio de Janeiro, 2007, v. 4662, pp. 578-591.

[10] J. Nielsen. "Heuristic evaluation". In Usability Inspection Methods (Eds. Nielsen and Mack), John Wiley & Sons, New York, 1994, 448 pages.

[11] N. M. C. Valentim, T. Conte, J. C. Maldonado. "Evaluating an Inspection Technique for Use Case Specifications Quantitative and Qualitative Analysis". In Proceedings of the 17th International Conference on Enterprise Information Systems, 2015, pp. 13-24.

[12] N. M. C. Valentim, T. Conte. "Improving a Usability Inspection Technique based on Quantitative and Qualitative Analysis" (in Portuguese). In Brazilian Symposium on Software Engineering, 2014, pp. 171-180.

[13] N. M. C. Valentim, T. S. da Silva, M. S. Silveira, T. Conte. "Comparative study between usability inspection techniques about activity diagrams" (in Portuguese). In Proceedings of the 12th Brazilian Symposium on Human Factors in Computing Systems, 2013, pp. 92-101.

[14] A. Fernandez, S. Abrahão, E. Insfran, M. Matera. "Further analysis on the validation of a usability inspection method for model-driven web development". In Proceedings of International symposium on Empirical software engineering and measurement, Lund, Sweden, 2012, pp. 153-156.

[15] G. M. Donahue. "Usability and the Bottom Line". In: Journal of IEEE Software, 2001, v. 18 (1), pp. 31-37.

[16] N. Juristo, A. Moreno, M. Sánchez, M. Baranauskas. "A Glass Box Design: Making the Impact of Usability on Software Development Visible". In Proceedings of the 11th International Conference on Human-Computer Interaction, Rio de Janeiro, 2007, v. 4663, pp. 541-554.

[17] X. Ferré, N. Juristo, H. Windl, L. Constabile. "Usability Basics for Software Developers". In Journal of IEEE Software, 2001, v. 18 (1), pp. 22-29.

[18] A. Seffah, E. Metzker. "The obstacles and myths of usability and software engineering". In Communications of the ACM - The Blogosphere, 2004, v. 47 (12), pp. 71–76.

[19] Valentim, N. M. C., Conte, T. "Technical Report: Version 3 of MIT 1", Report Number 005, 2015. Available at: http://uses.icomp.ufam.edu.br/attachments/article/42/RT-USES-2015-0005.pdf.

[20] C. Sauer, D. R. Jeffery, L. Land, P. Yetton. "The Effectiveness of Software Development Technical Review: A Behaviorally Motivated Program of Research". In IEEE Transactions on Software Engineering, 2000, v. 26 (1), pp. 1-14.

[21] S. Shapiro, M. Wilk. "An Analysis of Variance Test for Normality (Complete Samples)". In Biometrika, 1965, v.52, pp. 591-611.

[22] N. Juristo, A. M. Moreno. "Basics of Software Engineering Experimentation". Kluwer Academic Publishers, 2001, 420 pages

[23] H. B. Mann, D. R. Whitney. "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other". In Annais of Mathematical Statistics, 1947, v. 18, pp. 50-60.

[24] K. Hornbæk. "Dogmas in the assessment of usability evaluation methods". In Behaviour & Information Technology, 2010, v. 29 (1), pp. 97-111.

[25] F. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology". In MIS Quarterly, 1989, v. 13, n. 3, pp. 319 – 339.

[26] F. Lanubile, T. Mallardo, F. Calefato. "Tool support for Geographically Dispersed Inspection Teams". In Software Process Improvement and Practice, 2003, v. 8, pp, 217-231.

[27] R. Garland. "The Mid-Point on a Rating Scale: Is it Desirable?" In Marketing Bulletin 2, 1991, pp. 66-70.

[28] R. Johns. "One Size Doesn't Fit All: Selecting Response Scales For Attitude Items". In Journal of Elections, Public Opinion, and Parties, v. 15 (2), 2005, pp. 237-264.

[29] F. Calefato, F. Lanubile, P. Minervini. "Can Real-Time Machine Translation Overcome Language Barriers in Distributed Requirements Engineering?" In IEEE International Conference on Global Software Engineering, 2010, pp. 257-264.

[30] C. Wöhlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wessl. "Experimentation in software engineering: an introduction". Kluwer Academic Publishers, 2000, 236 pages.

[31] V. R. Basili, F. Shull, F. Lanubile, "Building knowledge through families of experiments". In IEEE Transactions on Software Engineering, 1999, v. 25, pp. 456-473.

[32] J. Carver, L. Jaccheri, S. Morasca, F. Shull. "Issues in Using Students in Empirical Studies in Software Engineering Education". In Proceedings of the 9th International Symposium on Software Metrics, Sydney, Australia, 2003, pp. 239-249.

[33] B. A. Kitchenham, S. Pfleeger, D. C. Hoaglin, K. El Emam, J. Rosenberg. "Preliminary guidelines for empirical research in software engineering". In IEEE Transactions on Software Engineering, 2002, v. 28 (8), pp. 721–734.

[34] R. H. Hartson, T. S. Andre, R. C. Williges. "Criteria for evaluating usability evaluation methods". In International Journal of Human–Computer Interaction, 2003, v. 15 (1), pp. 145–181.

[35] O. Laitenberger, C. Atkinson, M. Schlich, K. Emam. "An experimental comparison of reading techniques for defect detection in UML design documents". In Journal of Systems and Software, 2000, v. 53(2), pp. 183-204.