Evaluating Software Engineers' Acceptance of a Technique and Tool for Web Usability Inspection

Luis Rivero[1], Auri Vincenzi[2,3], José Carlos Maldonado[3] and Tayana Conte[1]
[1]USES Research Group, Instituto de Computação, Universidade Federal do Amazonas, Manaus – Brazil
[2]Instituto de Informática, Universidade Federal de Goiás, Goiânia – Brazil
[3]Departamento de Ciência da Computação, Universidade de São Paulo (USP), São Carlos – Brazil
{luisrivero, tayana}@icomp.ufam.edu.br; auri@inf.ufg.br; jcmaldon@icmc.usp.br

***Abstract*— Usability is related to software quality, improving its ability to be understood, operated and attractive to users. We proposed the Design Usability Evaluation (DUE) technologies to allow identifying usability problems earlier in the development of Web applications, through the inspection of mockups. While we found that the DUE technique and tool were effective and efficient in the identification of usability problems, we saw the need to investigate their acceptance in practitioners' work environment. This paper reports the results from a study evaluating the acceptance of the DUE technologies from the point of view of software engineers. We asked questions based on the indicators from the Technology Acceptance Model and identified that a majority of the software engineers who participated in the study: (a) found the DUE technologies useful and easy to use for supporting the usability inspection process; and (b) would regularly use the DUE technologies for future inspections in their job. Nevertheless, the practitioners indicated that the technique should be refined in order to reduce the ambiguity and repetition of some of its items, while the tool should become more intuitive.**

Keywords- Web usability; software quality; inspection technique; inspection tool; software testing tool; empirical study; technology acceptance

I. Introduction

A Web Application is a software system based on technologies and standards of the World Wide Web Consortium (W3C[1]) that provides Web specific resources such as content and services through a user interface, the Web browser [1]. Due to their importance for presenting products and services to customers, Web applications need to be usable so that they can be effective, efficient and satisfying to users [1]. In that context, usability subsumes aspects such as learnability, operability, aesthetics, and others that affect the quality of the developed applications [2]. Therefore, usability plays a central role in their acceptance and adoption [3].

Usability inspection is one of the ways for identifying usability problems, in which inspectors check the conformity of software artifacts against a set of usability standards [1]. However, although the number of usability inspection methods for evaluating Web applications has increased, only a short number of these methods can be applied earlier in development [4]. Methods applied later in the development process, or when the application is released, can increase the cost of correcting the identified problems since the source code of the application will have already been written [5]. Also, the difficulty in identifying usability problems increases if the inspectors are not guided through the evaluation process or if they do not have tool support for reducing cognitive effort while performing an inspection [4].

The positive reports on the use of mockups (sketches of how an application would look like after its development) to support several early software engineering activities [6] motivated us to develop a set of technologies for the usability inspection of mockups of Web applications [4]. These technologies, called Design Usability Evaluation (DUE), provide a technique and tool. While the technique provides a set of verification items that guide inspectors through the evaluation process, the tool facilitates its application by simulating interaction among the evaluated mockups and allowing pointing usability problems and generating reports.

In our previous work, we conducted empirical studies, showing indicators of the feasibility of the DUE technologies in the usability inspection of mockups in terms of effectiveness and efficiency in different conditions and when compared with other techniques [7][8]. However, these studies were carried out in academic environments or did not focus on the aspects that needed to be improved to enhance the acceptance of DUE the technologies in a real usage scenario. A good understanding of real software engineers' attitude towards the DUE technologies is expected to help us decide whether and how the technologies should be tailored to improve the results of the usability inspection of the mockups of Web applications.

According to Shull et al. [9], studies in a particular development lifecycle can help evaluating if new proposed software engineering technologies are compatible with software engineers' work environment. The results from these studies can reveal issues that did not arise during feasibility studies, allowing fine-tuning or tailoring of the technology to meet the needs of the software industry. This type of studies is essential for the industry in order to decide whether they will adopt or reject a specific technology. Therefore, following our evaluation of the feasibility of the DUE technologies in terms of effectiveness and efficiency [7], this paper presents a study in a real lifecycle. In this study, software engineers with experience in software verification and validation tried the DUE technologies and reported their perceived usefulness and perceived ease of use towards them. The goal of this paper is to

[1] http://www.w3.org/

DOI reference number: 10.18293/SEKE2015-114

report on their perception of the DUE technologies and their degree of acceptance. Additionally, we gathered data on what would make practitioners adopt or reject the DUE technologies.

II. Background and Motivation

According to Fernandez et al. [10], in order to assist the identification of usability problems in Web applications, new research has been performed in the field of usability evaluations. Such evaluations can range from [4]: (a) User Testing, in which users perform tasks so that an observational team can identify communication gaps and usability problems regarding the user interface; and (b) Usability Inspection Methods (UIMs), in which inspectors verify the conformity of software artifacts against a set of usability standards. The main advantage of applying inspection methods is that they require fewer resources to be applied. Since UIMs do not require special equipment or laboratories to be executed, they can lower the costs of the identification of usability problems [1].

In our previous work [4], we carried out an analysis over the review by Fernandez et al. [10], gathering data on UIMs for Web applications. Among the analyzed methods, Paganelli and Paterno [11] proposed a UIM that compares the way in which a Web system is expected to be used and the way in which it is really used, to identify usability problems. Also, Allen et al. [12] developed the Paper-Based Heuristic Evaluation, an inspection method evaluating mockups of medical Web applications in terms of consistency, minimalism, match, memory and language. Finally, Molina and Toval [13] proposed a method that provides a total of 50 metrics in order to identify usability problems from a meta-model formed by merging the navigational and requirements models.

Although the above methods provide means for identifying usability problems in Web applications, there is still room for improvement. For instance, methods such as the one proposed by Paganelli and Paterno [11] require that the source code of the application is available so users can experience it, which increases the cost of correcting the identified problems [5]. In fact, our analysis [4] showed that only around 15% of the identified usability evaluation methods could be applied at earlier stages of the development process. Moreover, from the methods that can be employed earlier in development (e.g. the Paper-Based Heuristic Evaluation), most of them do not provide guidance for software engineers applying them and/or do not provide tools to support inspectors in the evaluation of Web applications [4]. It is necessary to develop technologies (i.e. methods and tools) that address these issues in order to enhance the performance of software engineers in the evaluation of Web applications. The above needs have been considered in the proposal of the DUE technologies.

III. The DUE Technologies

We proposed the Design Usability Evaluation (DUE) technologies in order to meet the needs of inspection methods in the field of early usability evaluation of Web applications [4]. In this sense, the DUE technologies are a technique and a tool to guide software engineers in the identification process of usability problems in mockups of Web applications.

To guide inspectors in the identification of usability problems, the DUE technique suggests dividing mockups into Web page zones, which are pieces of a Web page that contain specific components to perform certain functionalities [4]. Examples of Web page zones are: the navigation zone, where the user can find means to go from one part of the application to another; the system state zone, where the user can find information of his/her location in the application, how (s)he got there and the available options in it; and others. Based on these zones, the DUE technique provides a set of verification items to check whether a usability problem can occur. For instance, Table I shows some of the verification items of the DUE technique for the data entry zone, while Fig. 1 shows a mockup in which these items have been violated. When an inspector verifies that there is a nonconformity, (s)he marks the item within the mockup, identifying the problem. As an example, problem P01 in Table I shows that in the data entry zone, the Web application does not provide hints for filling the fields, which can cause difficulties in inputting data (see Fig. 1 error A). Also, problem P02 in Table I shows that the fields that are mandatory are not highlighted, which can cause users to forget to provide relevant information (see Fig. 1 error B). Interested readers can find a complete description of the Web page zones and verification items from the DUE technique, and more examples of usability problems in our previous work [4].

TABLE I. A Web page zone and some of its verification items.

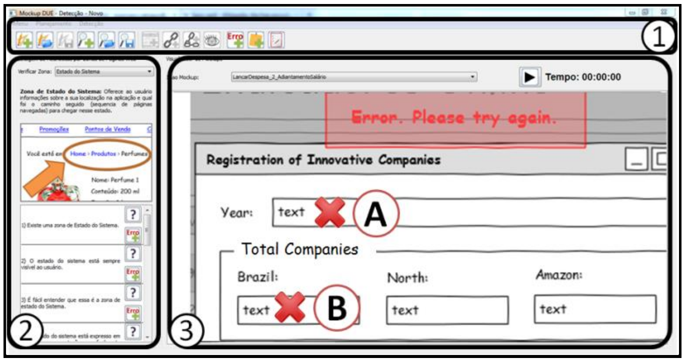
Data Entry Zone: This zone is responsible for providing the user with means of entering data into the application in order to allow the user to perform operations. Later, the user will click in a "submit" like button that will activate a function based on the entered data.	
ID	**Usability Verification Item**
P01	The interface indicates the correct format for a determined data entrance (e.g. a "Date" entry field could have the next hint: "mm/dd/yy").
P02	The interface indicates which data must be mandatory filled (e.g. mandatory input data is indicated with a "*" or a "mandatory" next to the field).

The initial evaluation of the DUE technique [7] showed that applying the technique on its own was tiring for inspectors, as they were forced to simulate the interaction between the user and the mockups. Therefore, we proposed the DUE tool to facilitate the application process of the DUE technique. To simulate navigation, the DUE tool allows inspectors to click on previously added links that, when activated, show the mockups in a sequence, to resemble a real application. Also, the DUE tool embeds the verification items from the DUE technique and shows them to the inspectors next to the set of evaluated mockups. This way, inspectors can request further information and details on the verification items while identifying usability problems. Fig. 1 shows the DUE tool when employed in the evaluation of a mockup. Using the tool, inspectors can point errors and notes (see area of the screen indicated by 1), view the verification items of the technique and simulate interaction (see areas of the screen indicated by 2 and 3 respectively).

Although there are other tools for creating mockups and simulating interaction (e.g. Mockingbird[2], Balsamiq Mockups[3] and others), these tools do not provide specific support for the usability inspection of the developed mockups. It is possible for

[2] https://gomockingbird.com/
[3] http://www.balsamiq.com/

inspectors to use these tools to simulate interaction, while identifying usability problems. However, it would be difficult to point the exact location of the encountered problems within the mockups while navigating through the application using these tools. In that context, the DUE tool allows inspectors to save the data on the inspection without using a spreadsheet, which would facilitate to pause and resume the inspection process whenever it is necessary. These are the main advantages of the DUE tool when compared to other tools.


Zona de Estado do Sistema: Oferece ao usuário informações sobre a sua localização na aplicação e qual foi o caminho seguido (sequencia de páginas navegadas) para chegar nesse estado.

1) Existe uma zona de Estado do Sistema.

2) O estado do sistema está sempre visível ao usuário.

3) É fácil entender que essa é a zona de estado do Sistema.

Error. Please try again.

Mockup DUE - Detecção - Novo
1
Erro
Verificar Zona: Estado do Sistema
Promoções
Pontos de Venda
Você está em Home › Produtos › Perfumes
Nome: Perfume 1
Conteúdo: 200 ml
?
Erro
2
Mockup: LancarDespesa_2_AdiantamentoSalário
Tempo: 00:00:00
Registration of Innovative Companies
Year:
text
A
Total Companies
Brazil:
North:
Amazon:
text
B
text
text
3

Figure 1. The DUE tool in its evaluation mode: (1) functionality bar, (2) verification items from the DUE technique, and (3) mockup being evaluated.

In order to complete an inspection using the DUE technologies, the inspector must evaluate all the verification items for all the zones present within the evaluated mockups. Then, (s)he can generate an automatic report containing information on the inspector and the identified problems. Later, such report will be discussed by the development team in order to verify which corrections are necessary and their priority.

IV. Empirical Evaluation

In this study, we aimed at: (a) assessing the acceptance of the DUE technologies by software engineers; and (b) identifying constrains and improvement opportunities for adopting these technologies in the software industry. To gather data for evaluating the acceptance of the DUE technologies, we applied a questionnaire based on the indicators from the Technology Acceptance Model (TAM). TAM [14] aims at assessing users' beliefs about the usefulness and ease of use of a technology that is expected to support their work. According to Davis [14], the reason for focusing on those indicators is that they are strongly correlated to user acceptance of a given technology.

The empirical study to assess the DUE technologies was conducted during a two-week professional training on software verification and validation. The goal of the course was to teach software engineering practitioners about new techniques and tools for guaranteeing quality in the software development lifecycle. One of the topics from the course was usability evaluations and our research team was asked to provide training on technologies for evaluating the usability of different software applications and their suitability in different stages of the development process. Thus, a training regarding the DUE technologies was prepared as part of the course.

Table II shows the questionnaire we applied for evaluating perceived usefulness and perceived ease of use on the DUE technologies. We based our questionnaire on the one by Laitenberger and Dreyer [15]. However, we selected only part of the items that could provide information on what could be improved in the development of the DUE technologies. In order to apply the questionnaire, we:

- Replaced the "technology" investigated in the questionnaire with the terms "DUE technique" or "DUE tool" according to the technology we were evaluating.

- Replaced the process investigated in the questionnaire with "usability inspection" with a focus on Web mockups.

- Employed a four-point scale asking for the degree of agreement with the statements from the point of view of software engineers: (1) Strongly Disagree, (2) Partially Disagree, (3) Partially Agree and (4) Strongly Agree. We did not use an intermediate level so the software engineers would provide information regarding the side to which they were inclined (either positive or negative) [15].

- For each of the statements within the questionnaire, we included open questions, asking for the reason why a subject chose a specific answer. This was done in order to better understand the features that made the DUE technologies useful (or useless), easy (or difficult) to use and suitable (unsuitable) for a software engineer's work.

At all, 20 software engineers from 5 different software companies (at Manaus-Brazil) were enrolled in the training and agreed to participate in the study. These software engineers had a strong technical background (knowledge in the planning, creation and documentation of test cases), and varying degrees of work experience in the testing of software applications (ranging from 2 to 10 years of experience – median 5 years).

The study took place in two days from the two-weeks training. Each day, the subjects entered a lab room where they had lectures and carried out real evaluations for a period of 4 hours with a 30 minutes break. During the first half of the training of the first day, the subjects received training in the application of different usability evaluation techniques (e.g. user testing [1], the heuristic evaluation [16]). After that, the subjects performed evaluations using these techniques over real applications under development. Then, during the second half of the course of the first day, the subjects were trained on the DUE technique, applied on its own without tool support, for inspecting the usability of the mockups of a Web application. Next, they performed the evaluation of a set of mockups and filled in the questionnaire statements regarding their acceptance of the DUE technique. During the second day, the subjects had training on the DUE tool. However, this time, they had to carry out an inspection over a real application under development comprising over 10 mockups. This was done in order to resemble a real evaluation scenario in industry and let the subjects experience the navigation functionalities from the DUE tool. Finally, all subjects filled in the questionnaire with statements regarding their acceptance of the DUE tool. We highlight that we did not compare the DUE technologies with other usability evaluation approaches as this study focused on the acceptance of the DUE technologies by software engineers. Studies comparing the DUE technologies to other approaches can be found in our previous work [7].

TABLE II. QUESTIONNAIRE STATEMENTS ON: PERCEIVED USEFULNESS, EASE OF USE AND FUTURE USE.

Statements regarding "Perceived Usefulness" (U):	
U1	Using the "technology" in my job would *improve my effectiveness* in a usability inspection of the mockups of a Web application.
U2	Using the "technology" in my job, I would be able to carry out a usability inspection of the mockups of a Web application *more quickly*.
U3	Using the "technology" in my job would *improve my performance* in a usability inspection of the mockups of a Web application.
U4	I would find the "technology" *useful* to carry out a usability inspection of the mockups of a Web application.
Statements regarding perceived "Ease of Use" (EoU):	
EoU1	*Learning to operate* the "technology" to carry out a usability inspection of the mockups of a Web application *would be easy* for me.
EoU2	I would find it easy to get the "technology" *to do what I want it to do* to carry out a usability inspection of the mockups of a Web application.
EoU3	It would be *easy to become skillful in using* a usability inspection technique/tool like the "Technology".
EoU4	I would find a usability inspection technique/tool like the "Technology" *easy to use*.
Statements regarding "Self-Predicted Future Use" (FU):	
FU1	Assuming a usability inspection technique/tool like the "Technology" would be available on my job, *I predict that I will use it* on a regular basis in the future.

V. DATA ANALYSIS AND RESULTS

Usefulness and Ease of Use are important measures for technology acceptance. We used the questionnaire to gather software engineers' opinion about their acceptance of the DUE technologies. Table III shows the descriptive statistics for the Usefulness statements (U1 to U4), Ease of Use statements (EoU1 to EoU4) and Self-Predicted Future Use. We have analyzed the results verifying the mean and standard deviation of the scores as in the examples by Laitenberger and Dreyer [15] and Babar et al. [17]. An average response between 3 (Partially Agree) and 4 (Strongly Agree) seems overall a positive result. The overall score for perceived usefulness and perceived ease of use has been calculated by summing the individual scores of their respective items, thus the maximum score is 16. Despite the cautiously positive results, some subjects were not convinced about the usefulness and ease of use of the DUE technologies. To better understand the reasons that made the subjects answer positively or negatively, we have analyzed the answers to the open questions.

Reasons that made software engineers believe that the DUE technique was useful were regarding the guidance and standards that were provided. For instance, one of the subjects indicated that since the technique focused on specific parts of the application and its attributes, it was easier to concentrate and identify the usability problems (see quote from Inspector I10). Furthermore, the verification items and their detailed description according to the zones that were being evaluated made software engineers believe that they would be able to find more problems (see quote from Inspector I04).

"(...) it allows me to focus on different areas and inspect them independently." – Inspector I10.

"*I believe it is effective as it supports identifying the defects through its well described items.*" – Inspector I04.

Overall, 5 inspectors disagreed with at least one of the items on the usefulness of the DUE technique. The main reasons for their disagreement was regarding the time it would take to carry out the inspection due to the large number of verification items (see quote from Inspector I08); and the overlapping between some of the items, which could confuse inspectors when looking for usability problems (see quote from Inspector I17).

"It can be tiring and take a long time depending on how many of the verification items you check." – Inspector I08.

"*The number of items and, in some cases, their ambiguity makes it diminish my performance and it takes time.*" – Inspector I17.

TABLE III. MEAN AND STD. DEV. FOR USEFULNESS, EASE OF USE AND SELF-PREDICTED FUTURE USE (FOUR-POINT SCALE: 1 TO 4).

	DUE Technique		DUE Tool	
Item	**Mean**	**Std. Dev.**	**Mean**	**Std. Dev.**
U1 - Effectiveness	3,70	0,46	3,58	0,49
U2 - Quick	3,15	0,79	3,50	0,65
U3 - Performance	3,60	0,49	3,50	0,50
U4 - Useful	3,40	0,73	3,08	0,86
Total Usefulness	**13,85**	**1,68**	**13,67**	**1,37**
EoU1 - Easy to learn	3,70	0,46	3,50	0,50
EoU2 - Controllable	3,50	0,67	3,33	0,62
EoU3 - Skillful	3,60	0,58	3,58	0,49
EoU4 - Easy to use	3,30	0,84	3,42	0,64
Total Ease of Use	**14,10**	**2,14**	**13,83**	**2,07**
Self-Predicted Future Use	**3,50**	**0,50**	**3,50**	**0,65**

Regarding the ease of use of the technique, the software engineers indicated that it was easy to identify usability problems as the technique pointed, for the different parts of the application, what an application should provide to be usable (see quote from Inspector I02). Furthermore, the software engineers indicated that the organization of the technique made it easier to learn and follow the inspection process (see quote from Inspector I10).

"It makes it easier since the zones and items make it clear what an application should provide, and what it actually has/lacks." – Inspector I02.

"Yes, using the zones makes it easier to follow the process and identify specific problems in which we would not focus in other circumstances." – Inspector I10.

Despite the positive feedback on ease of use, around 4 software engineers disagreed with at least one of the statements from the questionnaire regarding the DUE technique. Again, the main problem was the overlap between some of the verification items (see quote from Inspector I13). Moreover, other inspectors indicated that initially, the zones were not that intuitive, but as there were examples, one could learn how to use the technique (see quote from Inspector I18). We highlight that some inspectors suggested developing a tool support for the DUE technique (see quote from Inspector I14). While using the DUE technique, the software engineers participating in the study did not know that a tool was available. Thus, it can be an indicator supporting our results in our literature review and previous studies [4][7], which suggest that a tool is important for facilitating the use of UIMs.

"It was not that easy to use as some of the items are ambiguous or overlap, which makes it confusing and take more time" – Inspector I13.

"Initially, I had some difficulty in understanding the zones and their items. However, the examples made me overcome that problem, and I was able to apply it." – Inspector I18.

"I believe that the technique is suitable if the evaluation is short. However, in bigger applications, it would be better to have a tool to facilitate its use." – Inspector I14.

Regarding the DUE tool, and its use on the evaluation of the mockups of a real Web application under development, most software engineers provided positive feedback. When asked about the reasons that made the tool useful and easy to use, the inspectors indicated that the tool was useful as it made the inspection process more agile and quick (see quote from Inspector I13). Furthermore, the tool was perceived as intuitive and easy to use as the provided functionalities were easy to understand (see quote from Inspector I10).

"I believe it is a great tool, it makes the inspection process more agile and it makes it easier. It is an adequate and useful tool for the inspection." – Inspector I13.

"In my opinion, the tool was useful and it was easy to understand the provided options. Also, the way in which the errors are documented helped me. It is very intuitive." – Inspector I10.

At least 4 software engineers disagreed with one or more statements regarding the usefulness and ease of use of the DUE tool. These inspectors indicated that since the technique had many zones and verification items to be checked, finding them in the tool was also difficult and make using the tool inefficient (see quote from Inspector I08). Also, the appearance of the tool and the way it presented some feedback to the inspectors were not adequate in certain situations. For instance, Inspector I19 pointed out that the way in which the tool pointed the defects made it hard to visualize an application with many defects. In Fig.1 we can see that for each identified problem, the tool adds an "X" mark next to the problem (the inspector can relocate the X over the problem to make that problem easier to find in the report). When a mockup has many problems, as the number of marks increases, it turns difficult to view the mockup. Finally, the inspectors indicated that the inspection report should be reduced (see quote from Inspector I12), as it shows all the evaluated mockup, even if usability problems were not identified on them, thus wasting the time of the development team, when reviewing the reports.

"Since it presents all the zones and items from the technique, it is also tiring. Furthermore, it is difficult to identify previous problems that were added when identifying a usability problem." – Inspector I08.

"When we report a problem, the tool adds an 'X' to point it on the mockup. However, as the problems were being reported I was forced to relocate them so they would not make it difficult to navigate and view the mockups." – Inspector I19.

"I think that the report contains too much information. It could show the mockups in which problems have been found instead of showing all of them and wasting time." – Inspector I12.

When asked if they would employ the DUE technologies in their work environment, the majority of the subjects (strongly or partially) agreed that they would use it. However, only Inspector I19 disagreed with adopting the DUE tool in his/her job. The reason for this answer was that (s)he did not like the tool because of its design. Other inspectors indicated that the tool could be improved by grouping its functionalities (and buttons) according to their frequency of use, and providing shortcuts to make it faster to use. Also, they indicated the need for facilitating the navigation among the mockups and, perhaps, allowing importing mockups from other tools, instead of creating them elsewhere and mapping them into the tool. Finally, they indicated that in the first use, the tool should provide a quick introduction, so inspectors can be more familiar with its functionalities before starting the inspection. Regarding the DUE technique, the software engineers suggested creating generic items for those that were repeated in the zones. Also, they suggested making the ambiguous verification items more clear, by adding further information and hints on what a usable interface should provide.

VI. Limitations of the Study

Regarding the subjects' need for training, it would have been better if there was no need for it. However, the short training time allows the DUE technologies to be applied by software engineers with low experience in usability evaluations. In that context, the moderator and training could have caused an effect in the software engineers' perception of usefulness and ease of use. Nevertheless, the moderator did not highlight the (dis)advantages of the DUE technologies. Instead, he explained their application process and provided equivalent examples for all methods described in the training. Furthermore, when filling out the questionnaire, the moderator highlighted that the goal of the study was to identify improvement opportunities in the DUE technologies, encouraging the software engineers to be as honest as possible. Finally, besides the DUE technologies, the software engineers applied different usability evaluation techniques to guarantee that they could have a baseline to compare them. However, as the duration of the study depended on the duration of the training, we only gathered data on the acceptance of the DUE technologies.

Regarding the generalization of our findings, the representativeness of the inspected mockups can be a limitation. Although these mockups might not be representative of all types of applications [1] and inspectors may have different results evaluating other applications, these mockups were produced for a real system under development, resembling a real industrial usability evaluation scenario. Therefore, the results from this study must be considered indicators and further studies evaluating different types of applications should be executed. Also, since the number of subjects is low, the data extracted from this study can only be considered indicators and not conclusive. Nonetheless, it might not be possible to get sufficient size of data sets. Therefore, even with a small sample used, the results from this study are good indicators for explaining the reasons why users would accept or reject the DUE technologies.

A final limitation could be the instrument and measures applied in this study for assessing technology acceptance. However, we believe that applying questionnaires was more suitable than applying interviews due to time constrains. Furthermore, by evaluating perceived usefulness and perceived ease of use, we intended to have an idea of users' acceptance of the DUE technologies and identify issues that should be corrected to meet the needs of the software industry. Finally, the questions we asked to the software engineers were based on questionnaires applied in other researches [15][17] which have been previously validated.

VII. Conclusions and Future Work

We developed a set of usability inspection technologies for the evaluation of mockups of Web applications earlier in their development process. In this paper, we have studied the user acceptance of these technologies for carrying out usability evaluations. We used a questionnaire evaluating indicators based on the TAM model, for gaining understanding of the subjects' attitude towards the DUE technologies for inspecting the usability of mockups of Web applications. In that context, we found out that:

1. A majority of the subjects found the DUE technique and tool quite useful and easy to use for supporting the usability evaluation of mockups of Web applications.

2. Most of the software engineers who participated in the study would adopt the DUE technologies in their job.

3. The practitioners who disagreed with the statements from the questionnaire in terms of usefulness and ease of use indicated that to improve their performance, the DUE technique should reduce or combine some of its verification items and make them less ambiguous.

4. It is necessary to improve the design of the DUE tool to make it easier in its first usage experience. Also, the way in which problems are pointed should be improved so the visualization and navigation among the mockups are not affected.

As we had already evaluated the effectiveness and efficiency indicators of the DUE technologies in different contexts [7], this paper focused on the evaluation of their acceptance by software engineers. However, we still need to carry out further studies verifying to what extend previous knowledge on usability evaluations and previous practical experience affect the acceptance and performance of practitioners when applying the DUE technologies. Thus, as future work, we intend to replicate this study, but increasing the number of subjects, and analyzing their actual effectiveness and efficiency according to their experience. Also, in this new study, it is necessary to implement the changes suggested by the software engineers in order to improve the usefulness and ease of use of the DUE technologies and their adoption in the software industry. Furthermore, although we analyzed the answers to the open questions, we still need to carry out further qualitative analyses with other methods to better investigate the aspects that need to be improved to enhance their adoption.

Acknowledgments

We thank CNPq for the scholarship granted to the first author of this paper and for its financial support through process n° 460627/2014-7. Also, we thank the financial support granted by FAPEAM through processes nº: 01135/2011; 062.00146/2012; 062.00600/2014; 062.00578/2014; and PAPE 004/2015. Finally, we thank the financial support granted by FAPESP through processes nº 2014/15514-2.

References

[1] G. Kappel, B. Proll, S. Reich, W. Retschitzegger, "An Introduction to Web Engineering", In: G. Kappel, B. Proll, S. Reich, W. Retschitzegger, "Web Engineering: The Discipline of Systematic Development of Web Applications", John Wiley & Sons, pp. 1-17, 2006.

[2] International Organization for Standardization, ISO/IEC 25010, "Systems and software engineering -- SQuaRE - Software product Quality Requirements and Evaluation", 2011.

[3] L. Olsina, G. Covella G. Rossi, "Web Quality", In: E. Mendes, N. Mosley, Web Engineering, Springer, pp. 109-142, 2006.

[4] L. Rivero, R. Barreto, T. Conte, "Characterizing Usability Inspection Methods through the Analysis of a Systematic Mapping Study Extension", In Latin-american Center for Informatics Studies Electronic Journal, 16(1), pp. 12-12, 2013.

[5] R. Charette, "Why software fails", In IEEE Spectrum, 42(9), pp. 42-49, 2005.

[6] 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, 9(2), pp. 132-156, 2010.

[7] L. Rivero, T. Conte, "Improving Usability Inspection Technologies for Web Mockups through Empirical Studies", Proc. 25th SEKE, pp. 172-177, 2013.

[8] L. Rivero, M. Kalinowski, T. Conte, "Practical Findings from Applying Innovative Design Usability Evaluation Technologies for Mockups of Web Applications", Proc. 47th HICSS, pp. 3054-3063, 2014.

[9] F. Shull, J. Carver, G. Travassos, "An empirical methodology for introducing software processes", In ACM SIGSOFT Software Engineering Notes, 26(5), pp. 288-296, 2001.

[10] A. Fernandez, E. Insfran, S. Abrahão, "Usability evaluation methods for the web: A systematic mapping study", In Information and Software Technology, 53(8), pp. 789-817, 2011.

[11] L. Paganelli, F. Paterno, "Automatic reconstruction of the underlying interaction design of Web applications". Proc. 14th SEKE, pp. 439-445, 2002.

[12] M. Allen, L. Currie, S. Patel, J. Cimino, "Heuristic evaluation of paper-based Web pages: A simplified inspection usability methodology", In Journal of Biomedical Informatics, 39(4), pp. 412-423, 2006.

[13] F. Molina, A. Toval, "Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems", In Advances in Engineering Software, 40(12), pp. 1306-1317, 2009.

[14] F. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology", MIS quarterly, pp. 319-340, 1989.

[15] O. Laitenberger, H. Dreyer, "Evaluating the usefulness and the ease of use of a web-based inspection data collection tool", Proc. 5th International Software Metrics Symposium, pp. 122-132, 1998.

[16] J. Nielsen, "Finding usability problems through heuristic evaluation", Proc. CHI'92, pp. 373-380, 1992.

[17] M. Babar, D. Winkler, S. Biffl, "Evaluating the usefulness and ease of use of a groupware tool for the software architecture evaluation process", Proc. 1st ESEM, pp. 430-439, 2007.