

Practical Findings from Applying Innovative Design Usability Evaluation Technologies for Mockups of Web Applications

Luis Rivero
Universidade Federal do
Amazonas
luisrivero@icomp.ufam.edu.br

Marcos Kalinowski
Federal University of Juiz de
Fora
kalinowski@ice.ufjf.br

Tayana Conte
Universidade Federal do
Amazonas
tayana@icomp.ufam.edu.br

Abstract

Webflow is an optimal perceived user experience which improves the user orientation and navigational behaviors towards Web applications. In order for Web applications to achieve Webflow in terms of navigation, it is necessary to enhance their degree of user interaction. However, despite their importance, user interaction features that can improve ease of use and navigation are mostly evaluated late in development. This paper presents practical findings from applying proposed Design Usability Evaluation (DUE) technologies. Such technologies allow the usability inspection of Web applications mockups to reduce the cost of identifying and fixing navigation and ease of use problems early in the development process. We planned an empirical study to evaluate: (a) the capability of applying these technologies without training, and (b) the perceived ease of use and utility of the technologies, using the Technology Acceptance Model. The results show that the DUE technologies could be used without training to find problems affecting navigation and ease of use of Web applications. Also, practitioners agreed that using the DUE technologies could enhance their performance during an inspection. Finally, we provide a list of some of the identified user interaction problems that, if avoided, could aid in the improvement of Webflow.

1. Introduction

Our dependence and reliance on Web applications have significantly increased due to their importance in information exchange and business transactions [9]. In this sense, flow plays a central role in Web applications since it affects the user behavior towards them [12]. Flow is a state which occurs when navigating in an information space and which is intrinsically enjoyable, self-reinforcing and accompanied by a loss of self-consciousness [11].

In his work, Oinas-Kukkonen [12] modified the general flow concept into a Web-specific construct: “Webflow is an optimal perceived user experience which improves a web user’s orientation and navigational use, as well as vice versa, and which is predicted by balanced user skills and the feeling of the web to be enjoyably challenging, the feeling of being in control of web use, and the perceived ease of use and usefulness of the web”. In this sense, Webflow determines user engagement and is affected by the ease of use and degree of navigation of the Web applications user interface [11].

A Web application aiming at achieving Webflow should possess a high degree of user interaction, since it can improve its ease of use and navigational quality [5]. It is possible to verify all these features by carrying out user interaction inspections, in which expert evaluators, or even the development team, can examine design, navigational and ease of use related aspects of a user interface [14].

In order to assist software engineers in identifying navigational and ease of use problems early in the Web application development process, we proposed a set of usability inspection technologies for low fidelity prototypes (or mockups), which are images that show how the software would look like after its implementation [17]. These technologies are the Web Design Usability Evaluation (Web DUE) technique [16] and the Mockup Design Usability Evaluation (Mockup DUE) tool [15], which guide and assist inspectors in the identification of user interaction problems affecting user experience and engagement. Also, to facilitate identifying navigational problems, the Mockup DUE tool allows linking the mockups prior to the inspection so that their navigation can be simulated during the inspection. We have evaluated the DUE technologies through empirical studies to verify their feasibility [15]. The results from the studies showed that using the technique allowed finding more user interaction problems in less time, when compared to techniques that are used later in the development process [16].

In this paper, we extend our research regarding the evaluation and evolution of the DUE technologies when used by practitioners aiming at enhancing user interaction from early development stages. We have obtained industrial insights into their feasibility by conducting a new empirical study with an industrial scenario. In this scenario, practitioners with medium and high industrial software development experience have used the Web DUE technique through the Mockup DUE tool to carry out a user interaction inspection to detect navigation and ease of use related usability problems. The mockups that have been evaluated in the study were obtained from a functional software specification of a real development project and were linked together according to the provided specification before the actual inspection.

One of our goals with this study was to evaluate the industrial feasibility of applying the DUE technique by using the Mockup DUE tool without providing prior training, maintaining an acceptable result in the inspection when compared to subjects that received training. Thus, subjects were divided in two groups with different treatments. Also, qualitative data to gather industrial insights regarding the perceived usefulness, ease of use, and usage intention of the DUE technologies could be obtained by applying the Technology Acceptance Model (TAM) [22] and open questions. As a result of this evaluation we have identified improvement opportunities that could allow the use of the DUE technologies without training. Furthermore, we received positive feedback regarding the acceptance of both the technique and tool to help improving user navigation and ease of use of Web applications. Moreover, improvement suggestions that could help to safely transfer the DUE technologies from the academy to industry were obtained. Finally, we have gathered user interaction problems that could be avoided by software developers to enhance Webflow in terms of navigation and ease of use.

This paper is organized as follows. In Section 2, we describe related research and the motivation for creating the DUE technologies. In Section 3, we describe the current versions of the Web DUE technique and the Mockup DUE tool. In that section, we also provide a summary of the results from the first empirical studies evaluating the DUE technologies and the need for further investigations. In Section 4, we describe the planning and execution of the new empirical study with software practitioners focusing on navigational and ease of use problems. Then, in Section 5 we present the quantitative and qualitative results with their respective analyses, and a discussion of identified problems and their possible effect on Webflow. Finally, in Section 6 we provide our conclusions and future work.

2. Related Work and Motivation

Ease of use and navigational problems can be identified by performing evaluations which collect usage data related to end-user interaction with a software product. In this sense, inspection methods are a type of evaluation in which inspectors (or the development team) review the conformance of software artifacts with a set of guidelines [5]. Such guidelines can range from checking the level of achievement of specific user interaction and navigational attributes, to heuristics that predict user interface related problems.

Inspection methods evaluating user experience are gaining popularity due to their lower cost since they do not require any special equipment or laboratory to be performed [5]. Therefore, many inspection methods for guaranteeing user engagement and navigation quality in Web applications have been proposed [17]. In our previous work [14], we identified that despite their increasing number, most of these methods can only be applied late in the development process, mainly in the testing and deployment activities. The main drawback of evaluating user experience at those stages, is that the source code of the application will have already been written, increasing the cost of correcting any encountered user interaction problem [21]. Thus, the academy and industry have proposed specific inspection methods that can be used to identify user interaction problems in the beginning of the development process of Web applications [14].

Signore [19] proposed the Comprehensive Model for Web Sites Quality, which uses HTML and model analysis. Also, it uses five perspectives in order to evaluate the design of Web applications: correctness, presentation, content, navigation and interaction. During the HTML analysis, a tool verifies correctness problems. During the model analysis, inspectors relate correctness problems to the other perspectives and identify user interaction problems. The lessons learnt from each inspection are kept in a repository for further analysis.

Allen et al. [1] present the Paper-Based Heuristic Evaluation, which was designed for assessing the degree of quality design of medical Web applications mockups. During the inspection process, the inspectors evaluate mockups or Web application's print screens using a set of user interaction heuristics.

Despite the effort in evaluating the user interaction, navigability and engagement of Web applications early in the development, results from our previous work [14] showed that most of the proposed methods still required at least part of the application's source code in order to perform the evaluation. Furthermore, some of the methods can only be used to evaluate specific types

of Web applications like medical, transactional and Web 2.0 [14]. Also, the proposed tools did not assist inspectors in the identification of interaction problems [17].

When analyzing the current state of inspection methods evaluating the ease of use and navigation of Web application, we noticed that we needed to address some features to fully support the identification of user interaction problems at a lower cost [17]:

- **Feature 1:** Allow the identification of ease of use and navigational problems in early stages.
- **Feature 2:** Assist inspectors in identifying more Web application user interaction problems.
- **Feature 3:** Provide tool support for inspectors to enhance the inspection results.

In the following section we explain our solution, the DUE technologies, and how we met each of these previously mentioned features in order to provide a solution for identifying navigational and ease of use problems in early stages of the development process.

3. The DUE Technologies

The Design Usability Evaluation technologies were proposed in order to meet the software industry's needs regarding Web application inspection methods for evaluating usability in terms of navigation and ease of use. They are composed of a technique to evaluate the design and user interaction of mockups and a tool to support its inspection process. In the following subsections we present an overview of each of the technologies, how to apply them and their initial evaluations through empirical studies.

3.1. The Web DUE Technique

The Web Design Usability Evaluation (Web DUE) technique is an evaluation technique based on checklists that allows the evaluation of mockups of Web applications [16]. Mockups are a software model that uses images of how the software would look like after its implementation to document and validate the overall software design. Since mockups have a low creation cost, their popularity has risen [13]. Thus, by evaluating Web application mockups, the Web DUE technique assists the identification of ease of use and navigational problems in early stages (see Section 2 - Feature 1). Moreover, since mockups are created before developing the applications, the problems can be identified before the source code is written, lowering their correction cost [21].

Another feature of the Web DUE technique is that it guides the inspectors through the inspection process by using Web page zones and checklists (see Section 2 – Feature 2). A Web page zone is a piece of a Web

page with specific contents [6]. The Web DUE technique uses up to ten Web page zones, such as the navigation, system state, data entry, among others, to draw attention to specific user interaction and ease of use problems that a Web application can present. To identify such problems, each of these zones provides a set of verification items to be checked. For instance, Table 1 contains some of the verification items for the data entry zone, which provides users with a data input form to execute certain operations in the Web application. The complete list of Web page zones and their verification items can be found elsewhere [17][14].

Table 1. Verification items for the data entry zone.

Data Entry Zone		
Description: This zone is responsible for providing the user with a form to input data in order to carry out certain operations. Then, a submit-like button links the input data with the associated functionality.		
ID	Verification Items	Examples / Explanations
A	The interface indicates the correct format for a determined data entrance.	The data fields must contain/provide hints on how to fill them. For example, a "Date" entry field could have the next hint: "mm/dd/yy".
B	The interface indicates which data must be mandatory filled.	For example, the system indicates mandatory input data with a "*" or a "mandatory" next to the field.

To carry out a user interaction inspection on Web application mockups, inspectors must follow the steps illustrated in Figure 1. First, the inspector must identify the mockup's Web page zones. For instance, in the mockup of a financial module of a real Enterprise Resource Planning (ERP) system shown in Figure 1, Element 1 represents the institutional zone, which contains information about the authors or company responsible for the Web application. Element 2 represents two zones, the navigation and system state zone, which inform the location of the users in the application and where they can go. Finally, Element 3 shows the data entry zone, which is described in Table 1. It is noteworthy that the first step of the process must be done mentally by the inspectors, where they must judge which zones are contained in the mockups.

After identifying the zones, the inspector must check the verification items for each encountered zone. Any non-conformity with any of the items must be analyzed to predict whether there will be a user interaction problem. For instance, when evaluating Element B (see Table 1) in the data entry zone in Figure 1 (see Element 3), we can see that there is a non-conformity. The system does not inform the user which data is mandatory. Since this can lead users to make errors by leaving necessary information uninformed, there will be a user interaction problem.

All encountered user interaction problems must be described and marked in the mockup (see Figure 1).

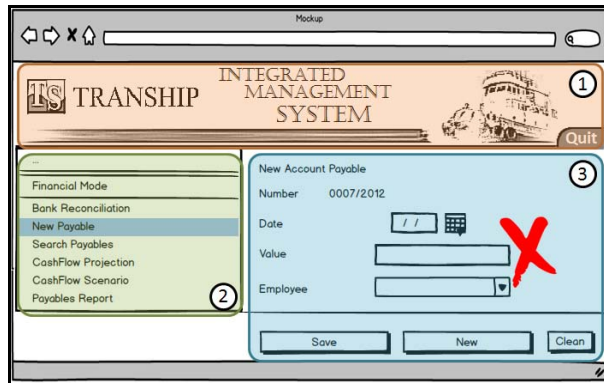


Figure 1. Web DUE technique - Inspection Process.

3.2. The Mockup DUE Tool

The Mockup Design Usability Evaluation (Mockup DUE) tool was developed in order to assist inspectors in applying the Web DUE technique. By using the Mockup DUE tool, inspectors can save time by not having to simulate the navigation between the different mockups of the application. Furthermore, since the Web DUE technique is embedded in the tool, the inspectors can easily switch between web page zones and the lists of verification items, which can enhance their performance (see Section 2 – Feature 3) [15].

The Mockup DUE tool supports two main activities: planning and inspection. During the planning stage, the moderators load mockups and link them so that inspectors can use the Mockup DUE tool to carry out the inspection. Then, during the inspection, the inspectors use the Mockup DUE tool to inspect a set of previously mapped mockups using the Web DUE technique. Figure 2 shows the layout of the Mockup DUE tool when it is used to carry out an inspection. In Figure 2, Element 1 shows the different functionalities of the tool: point problems, add notes and create reports. Furthermore, inspectors can view the Web DUE technique in Element 2. In this part of the screen inspectors can also read a description for each Web page zone and find explanations/examples for each verification item (see an explanation/example in Table 1). Finally, the mockups are shown in Element 3 according to the inspector's actions. This means, that if the inspector clicks in a link within the mockup, the tool will load the subsequent mockup, simulating navigation. Also, inspectors can describe and point the identified problems as shown in Figure 2 – Element 3.

One of the advantages of the tool is its ability to create complete inspection reports. Figure 3 shows an example of such reports containing information about the inspector, the list of user interaction problems and

notes, and the mockups with the exact location of the identified problems. These reports can be used by the development team to discuss changes in the design of the application and validate them with users.

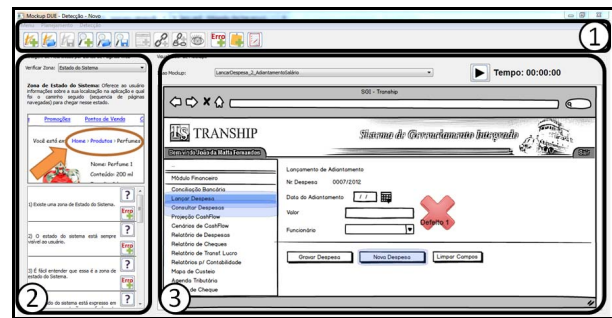


Figure 2. Print screen of the Mockup DUE tool.

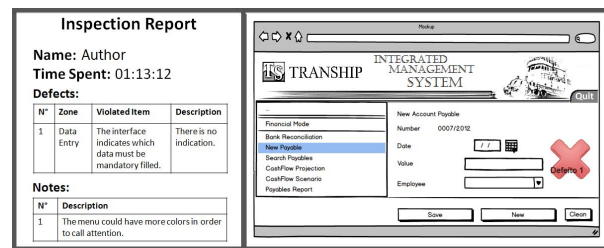


Figure 3. Mockup DUE tool – Inspection Report.

3.3. The Initial Evaluations of the DUE Technologies

The DUE technologies have been evaluated following a research methodology based on empirical studies [18]. We performed two empirical studies in which we evaluated the feasibility of the Web DUE technique by comparing it with other techniques that could be applied later in the development process, such as the Heuristic Evaluation [10] and the Web Design Perspective Based [16]. The results from these studies showed that the Web DUE technique was able to assist inspectors in finding more user interaction problems in Web application mockups in less time [16]. Also, we evaluated the feasibility of the Mockup DUE tool when used to perform a user interaction inspection with navigable mockups [15]. The results showed that inspectors found the tool adequate and easy to use.

However, the studies conducted so far had been performed in an academic context and, according to Shull et al. [18], in order to safely transfer a technology from academy to industry, an industrial study must be performed. Such study may provide important additional insights into unforeseen positive/negative aspects for practitioners working in real industrial settings. The study presented in this paper was planned and conducted in order to bridge this gap and its results are discussed hereafter.

4. The Empirical Study with Software Development Practitioners

We aim to extend our research in the evaluation, improvement and transfer from the academy to the software industry of the DUE technologies by providing industrial insights and answering the following research question: “*Are there any constraints in adopting the DUE technologies in an industrial environment?*”. In the following subsections we provide information needed by readers to assess the quality of the empirical study and the applicability of its results. Also, we have planned this study following the suggestions proposed by Wohlin et al. [23] for conducting empirical studies.

4.1. Goal

First, we want to identify if training is necessary to introduce the DUE technologies into the industry. Second, we want to predict the usage of the technologies by software development practitioners. Finally, we want to identify any further improvement opportunities and additional industrial insights, from the point of view of practitioners, to enhance the technologies and their results.

We have evaluated the effectiveness and efficiency indicators of the DUE technologies when applied with and without training. Such indicators are important because they provide information about the capability of an inspection technique to assist inspectors during the identification of problems [4]. For these indicators, we have used the same definition used in other researches [4][16]:

Effectiveness is the ratio between the number of detected defects and the total of existing defects.

Efficiency is the ratio between the number of detected defects and the time spent in finding them.

Regarding the feasibility of the technologies in terms of the industrial insights into usage prediction, and the constraints for their application, we have performed an evaluation using the Technology Acceptance Model (TAM) [22] evaluating the perceived usefulness, ease of use and behavioral intention of use from the point of view of practitioners. Also, open questions were applied to gather the practitioners’ opinions.

4.2. Hypotheses

Using the indicators defined above, we planned and conducted the study to test the following hypotheses (null and alternative, respectively):

H₀₁: There is no difference in terms of effectiveness in using the DUE technologies to find user interaction problems with or without training.

H_{A1}: The Web DUE technologies present a difference in the effectiveness indicator, when applied without training.

H₀₂: There is no difference in terms of efficiency in using the DUE technologies to find user interaction problems with or without training.

H_{A2}: The Web DUE technologies present a difference in the efficiency indicator, when applied without training.

4.3. Subjects

The empirical study was carried out in 2013, with 17 volunteer software practitioners working at different software development companies located around the Southeast area of Brazil. All subjects signed a consent form and filled out a characterization form. The characterization form addressed the subjects’ expertise concerning: (a) their educational level; (b) their experience in software development; (c) their expertise in Web applications development and design; (d) their expertise in user experience and user interaction inspections; and (e) their knowledge of the problem domain. All subjects answered objective questions regarding their degree of knowledge and professional experience. The characterization data was analyzed and each subject was classified as having Low, Medium or High experience according to the provided information. For instance, regarding the educational level, a subject was classified as having: (a) Low background if he/she had not finished college, (b) Medium background if he/she finished college, or (c) High if he/she holds a Masters or PhD degree. Similarly, the expertise in development, user interaction evaluation and the problem domain was assigned according to the number of years in which the subject had worked in such activities: (a) Low: less than 1 year; (b) Medium: 1 to 3 years; and (c) High: more than 3 years.

In order to reduce the bias of having more experienced inspectors in one or another treatment (with or without training), we equally distributed the subjects, balancing both teams’ subjects. Table 2 shows both teams and the expertise of their members.

There were other participants in this empirical study: (a) the inspection moderator, who was responsible for planning and collecting the data during the empirical study; (b) the team instructor, who provided the training on the technologies; and (c) the discrimination team, which was responsible for deciding which of the encountered defects were real user interaction problems.

4.4. Materials and Procedures

All subjects had trainings on: (a) how to develop mockups, (b) examples of typical user interaction problems, and (c) applying user interaction evaluations in general. Then, the task given was to perform a user interaction inspection on mockups of a real Web application under development.

The Web application was an ERP system (see a mockup in Figure 1) designed to meet the specific business needs of a medium sized (400 employees) naval company providing marine support services. The application included several modules (administration, operations and services, crew allocation, financial, human resources, among others). At all, so far more than 90 use cases were implemented resulting in an ERP system with 82,931 lines of Java code manipulating 167 tables. We found that this application would be a good object for our study, since besides the context of a real development project it represents a category of Web applications that according to Kappel et al. [7] should allow users to efficiently, consistently and interactively handle large amounts of data on the Web. Thus, a high degree of usability was strongly desired and ease of use and navigational problems should be avoided. Moreover, the functional specifications of the modules already included mockups used for customer validation before implementation. Thus, we decided to select the mockups of two use cases (10 mockups) regarding account payables of the financial module (a more common domain) for the user interaction inspection by the practitioners using the DUE technologies. The use cases were related to registering a new payable and searching for payables in order to quit them.

In order for the subjects to carry out the user interaction evaluation using the DUE technologies, the following instruments were provided: (a) the task instructions, which contained information on what artifacts should be delivered; (b) the documentation of the DUE technologies, containing the steps for performing an user interaction evaluation with the technique and tool; (c) the Mockup DUE tool executable file, which embeds the Web DUE technique; (d) the 10 Web application mockups with their navigation previously mapped; (e) the inspection guide, containing a description of the functionalities to be evaluated; and (f) the follow-up questionnaire asking for the subjects opinion on the DUE technologies and their usage intention.

After providing the materials for inspection the instructor indicated the subjects which group they belonged to. As mentioned before the subjects were equally distributed into two teams according to the characterization data delivered before the experiment.

Then, the instructor dismissed the group that would not participate in the training of the DUE technologies. The group that remained with the instructor received training regarding the same information provided in the documentation of the DUE technologies to avoid bias.

The context of the inspection comprised the mockups of the two selected use cases: (a) registering new payables and (b) searching for payables in order to quit them. Each subject had five days to complete the inspection. After finishing the inspection, each subject sent his/her report, which was automatically created with the Mockup DUE tool, and the follow-up questionnaire with comments regarding the DUE technologies.

4.5. Data Collection

All inspections reports were delivered on time and none of them were discarded. One of the authors who were responsible for conducting the empirical study was acting as the inspection's moderator. The moderator checked all discrepancies within the defect report for incorrect or missing information and also gathered the discrepancies. During the collection activity, the moderator highlighted duplicated discrepancies. Finally he generated a new discrepancies report which contained all discrepancies found without showing the duplicated ones.

After the collection activity there was a discrimination meeting to classify the discrepancies within the report as real defects or false-positives. This meeting was attended by the moderator and two other researchers who were not involved with the study. These researchers possessed good usability knowledge and prior experience in user interaction evaluation. For each discrepancy reported, the other researchers verified if it was a user interaction problem by evaluating the paper based prototypes. Also, the moderator did not classify any of the discrepancies in order to reduce bias. Considering all discrepancies, there were a total of 189 real defects. This number was use in the calculation of the effectiveness indicator.

5. Results and Discussion

We have analyzed quantitative data to test the hypotheses about the effect of training and qualitative data to understand any constraints in adopting the DUE technologies. The quantitative data was obtained by analyzing the time and number of defects per inspector and treatment, while the qualitative data was collected through the follow-up questionnaires. Furthermore, we present a list of some of the identified user interaction problems and their effect on Webflow.

5.1. Quantitative Analysis

Table 2 shows the overall results per inspector and technique. To calculate the effectiveness indicator we have used 189, which is the total number of defects. We have performed a statistical analysis using the SPSS v 20.0.0 tool [20]. In this analysis, we used $\alpha = 0.05$ due to the small sample used within this study [3]. In order to compare the effectiveness and efficiency of both samples, we used the Mann-Whitney non parametrical statistic method and the boxplots graph to facilitate visualization.

Table 2. Overall results per subject and treatment.

Team 1: Using DUE technologies without training									
Inspector ID	01	02	03	04	05	06	07	08	09
Educational Background	H	H	M	M	M	M	M	M	M
Exp. in SW Development	H	H	H	H	M	M	M	L	L
Exp. in Web Development	H	H	M	L	M	L	L	L	L
Exp. in User Interaction Evaluation	M	M	M	L	H	M	L	M	M
Exp. in Problem Domain	H	L	H	M	H	L	M	L	L
Discrepancies	24	27	19	6	87	15	29	69	29
False Positive	3	5	6	1	45	2	5	9	13
Total Defects	21	22	9	5	42	13	24	60	16
Time (min)	86	48	85	37	68	100	81	244	66
Effectiveness (%)	11,1	11,6	4,8	2,6	22,2	6,9	12,7	31,7	8,5
Efficiency (Defects/hour)	14,7	27,5	6,4	8,1	37,1	7,8	17,8	14,8	14,5
Average Effectiveness	12,5								
Average Efficiency	16,5								
Team 2: Using DUE technologies with training									
Inspector ID	10	11	12	13	14	15	16	17	-
Educational Background	H	M	M	M	M	M	M	M	-
Exp. in SW Development	H	H	H	H	H	M	M	M	-
Exp. in Web Development	H	H	H	L	L	M	M	L	-
Exp. in User Interaction Evaluation	H	M	M	M	L	M	L	L	-
Exp. in Problem Domain	M	M	L	L	M	M	M	H	-
Discrepancies	23	57	19	44	21	47	30	20	-
False Positive	8	5	2	10	2	11	5	3	-
Total Defects	15	52	17	34	19	36	25	17	-
Time (min)	43	151	88	73	47	125	80	40	-
Effectiveness (%)	7,9	27,5	9,0	18,0	10,1	19,0	13,2	9,0	-
Efficiency (Defects/hour)	20,9	20,7	11,6	27,9	24,3	17,3	18,8	25,5	-
Average Effectiveness	14,2								
Average Efficiency	20,9								

Figure 4 shows the boxplots graph for the effectiveness and efficiency indicators. The results from the statistical analysis support both null hypotheses H_{01} and H_{02} indicating that there is no

significant difference in the effectiveness indicator when using the DUE technologies with or without training ($p = 0,481$); and that there is no significant difference in the efficiency indicator when using the DUE technologies with or without training ($p = 0,139$). Moreover, the graph in Figure 4 shows that, for effectiveness, the median is similar in both treatments, which suggests that perhaps, the documentation for using the technologies is sufficient to enable their use by the software development industry. However, we can argue that given the small sample used, it is difficult to obtain statistical significance. Thus, the results should be considered indicators and not conclusive. In fact, Figure 4 shows that the group that received training obtained slightly higher results in the efficiency indicator. We will explain the difficulty met by the subjects who did not receive training in the next subsection.

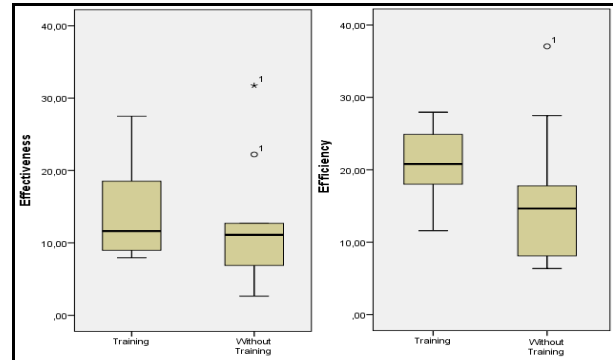


Figure 4. Boxplots graph evaluating the indicators.

5.2. Qualitative Analysis

The follow-up questionnaire was formulated to gather two types of information: (a) usage intention of the DUE technologies and (b) the constrains for using them, or improvement opportunities. It is noteworthy that we did not carry out interviews since they tend to cause the subjects to restrain vital information, especially about the disadvantages of the technologies. Thus, by using questionnaires we expect to allow subjects to express their feelings more freely. Furthermore, since some of the subjects were located in different places, it would be an easier approach to use questionnaires to obtain qualitative data.

As for the first type of data, we used the answers to the questions of the first part of the questionnaire which were based on the Technology Acceptance Model – TAM [22]. The TAM measures the acceptance and usage intention of new technologies by asking questions about the perceived usefulness, ease of use and behavioral intention of use [22]. For this part of the questionnaire, the answers were provided in a four level Likert Scale: strongly disagree, partially disagree, partially agree and strongly agree. We did not

use a five level Likert scale containing an intermediate level as suggested by Laitenberger and Dreyer [8] since this neutral level does not provide information regarding the side to which the subjects are inclined (either positive or negative).

In order to understand the answers to the questions based on the TAM model, we also included open questions in the follow-up questionnaire. Such questions came after the Likert scales asking for the reason why the subjects had selected a specific answer. For instance, regarding if the technique or tool would enhance the performance of the subject when performing an inspection, the subjects were asked what were the difficulties/advantages that could make the technologies inappropriate/suitable for the inspection of mockups of Web applications. In this sense, similar questions were made for the other statements used in the TAM model. In the next paragraphs we relate the answers to the TAM questionnaire and the overall results of the inspectors to the open questions.

Regarding the perceived usefulness of the Web DUE technique, there were only two inspectors who disagreed with some statements. Inspector 11 strongly disagreed that the technique would enhance his productivity. When looking at the quantitative data we can see that this inspector got the highest number of defects. However, he also took the highest amount of time. When asked about what affected his decision he answered that the number of verification items made it difficult to point a problem (see quote from inspector 11 below). Also, Inspector 05 partially disagreed that the technique would be useful. If we look at his results in Table 2 we can see that he managed to identify 42 defects. However, he also pointed 45 false positives. Since Inspector 05 did not receive training he argued that the items were ambiguous and it was difficult to identify user interaction problems (see quote from Inspector 05 below). Also, we noticed he did not understand the inspection process correctly. The nonconformity of a verification item does not imply in a problem, it is the inspector who must decide if it affects the user interaction of the Web application [15]. However, Inspector 05 pointed every nonconformity as a problem, even if a non mandatory zone was not present in the mockup, or if the precondition to evaluate the item was not met. This issue was observed in the results of other inspectors who did not receive training as well: 04, 08, 09. Therefore, we intend to improve the documentation with better descriptions of the inspection process to avoid future similar mistakes.

"... It was difficult to know how to relate the identified defects due to the high number of verification items" - Inspector 11.

"... It was difficult to identify the defects because some items weren't clear" - Inspector 05.

As for the perceived ease of use of the technique, we received positive feedback and only two inspectors disagreed with the statements. Inspector 11 found the technique difficult to use for the user interaction evaluation of mockups. Furthermore, Inspectors 11 and 15 found that it would be difficult to become skillful in using the Web DUE technique. When looking at the results from these inspectors, we can see that they achieved the better effectiveness results. However, the training they received in common user interaction problems could have had an effect in how easy it would be to carry out an inspection. For instance, Inspector 11 stated that it would be easy to learn how to use and become skillful in the Web DUE technique, but that would only be possible if previous knowledge in other techniques was acquired (see quote from Inspector 11 below). Furthermore, Inspector 15 argued that it was difficult to use the technique because sometimes the Web page zones can overlap (see quote from Inspector 15 below). In order to achieve better results, we intend to improve the documentation for the DUE technologies by providing an introduction to general user interaction problems and further information regarding how to treat a complex Web page zone that might be composed of different zones.

"... It would be possible to use the Web DUE technique without training if the inspector had prior usability knowledge and experience in user interaction evaluation. But without that knowledge I doubt it." - Inspector 11.

"... Identifying overlapping zones would make it difficult" - Inspector 15.

Regarding the perceived usefulness of the Mockup DUE tool most inspectors agreed that the tool is useful. However, Inspectors 2 and 8 partially disagreed that the tool would enhance their productivity. Furthermore, Inspectors 2 and 5 partially disagreed that the tool would be useful. As we can see in Table 2 none of these inspectors were trained. Consequently, they did not received information about the motivation of using the tool for inspection. Inspector 02 for instance, argued that the documentation lacked a motivation or goal for the tool. Furthermore, he stated that since the tool can only run in Windows, it could be inappropriate for his work environment (see quote from inspector 02 below). Moreover, although Inspectors 05 and 08 found the highest number of defects, they were not satisfied since they did not know if they were doing the correct activities in the tool. Moreover, as mentioned before, both Inspector 05 and 08 had trouble following the correct inspection process, therefore affecting their opinion towards the Mockup DUE tool.

“... In my case, since I am an OSX and Ubuntu user, it was inappropriate since I had to install Windows in order to use it” - Inspector 02.

As for the ease of use of the tool, we received positive feedback and only Inspector 02 partially disagreed that it would be easy to become skillful or use the Mockup DUE tool. This inspector found the instructions in the documentation somehow difficult to understand because there were many preconditions. He stated that it would be easier if there was a textual and more elaborated description. Also, he thought that the system state messages were good, but once knowing what was happening, users should be able to disable them. Moreover, he indicated that having to roll up to the top of the screen every time he added a new problem/note made it difficult to point user interaction problems (see quote from Inspector 02 below). We intend to make corrections to solve these problems.

“... I didn’t like that I had to go to the top of the page every time I wanted to point a problem...” - Inspector 02.

Finally, although most inspectors agreed they would use the DUE technologies, there was only one inspector who partially disagreed. Inspector 05 did not intend to use the technologies, perhaps due to the difficulty he had when trying to understand their use through the documentation. As mentioned before, he/she pointed every nonconformity as a problem and as a result had the highest number of false positives.

5.3. Identified Defects Affecting Webflow

Table 3 shows a small sample of the set of defects related to problems that affect ease of use and navigation within the evaluated mockups of the Web application. The first two problems (items 1 and 2) refer to navigational problems. For instance, it is important that developers provide meaningful names to their buttons and links, otherwise users might feel lost while navigating through the system. Furthermore, it is important to provide advanced options for high experienced users. Such problems are corroborated by the Webflow model in which ease of use is affected by the users’ skills [11]. Therefore, in order to provide a high quality of navigation, it is important to provide shortcuts for both experienced and novice users.

As sample of the identified problems affecting ease of use of Web applications, we listed items 3 to 6 (see Table 3). Thus, the DUE technologies allowed inspectors to identify which key user interface elements (buttons, links, others) needed to be renamed or provide a hint for them to be easy to use. Inspectors noted the need for hints in order to fill in data. Also,

when an application has alternative data to be input, this should be clear to the user. In the evaluated ERP application, several alternative paths were presented at once, which could make users believe that they had to fill in every field present in the interface rather than choosing a subset of them. Moreover, the system response is essential in order to inform the user of which actions have been performed successfully. Finally, in order to enhance Webflow, a Web application should provide easier ways to perform a task, by providing accelerator functionalities as pointed out in item 6.

Table 3. Identified user interaction problems.

ID	Verification Items	Examples / Explanations
1	The items in the menu are not standardized since some of them use verbs to explain their action while others use nouns.	
2	There is no direct access zone.	
3	There is no information on how to fill in the requested data.	
4	There is no help in the system. Some items are not familiar and without help it is difficult to know what they refer to.	
5	It is not possible to recognize whether an e-mail has been attached.	
6	Instead of selecting a worker by looking for it in a list, it would be much easier to allow a search by his name or id.	

6. Conclusions and Future Work

In this paper we extended our research on the DUE technologies by performing an empirical study with real mockups from a real ERP Web application being evaluated by practitioners. The quantitative results allowed us to identify that the technologies could be used with and without training, not seriously affecting the results of the inspection. However, in order to achieve the same results without training, the documentation for the DUE technologies must be improved. Also, analyzing the results from the evaluation using TAM, we received positive feedback regarding the utility, ease of use and intention of use of the technologies by practitioners.

We have discussed some of the identified user interaction problems, explaining their relationship with Webflow in terms of high quality navigation and ease of use. Also, it is important to point out that by performing inspections in Web application mockups, the software industry will be able to substantially reduce the cost of the identification and correction of the encountered problems. As stated by previous researches [2], by performing inspections early in the development, the cost in hours of finding user interaction problems can be reduced in more than 50% if compared to inspections performed in later stages. Furthermore, when compared to software testing, inspections can improve the productivity of fixing problems by reducing the time needed to correct them.

Thus, by using the DUE technologies, business companies can improve Webflow, enhancing user orientation and navigational behaviors [11] in Web applications at a lower cost.

As future work, we will improve the documentation of the DUE technologies by providing more details, explaining the inspection process and including the motivation and goals of the inspection. Moreover, although real software development practitioners participated in this study, we still need to increase the sample used to achieve more significant results from a statistical point of view. Thus, we will perform new empirical studies increasing the number of subjects. Also, we intend to perform further empirical studies in order to analyze which features influence the effectiveness and efficiency indicators of an inspection performed by practitioner using the DUE technologies.

7. Acknowledgements

We would like to thank FAPEAM and CNPq for their constant financial support regarding this research.

8. References

- [1] Allen, M., Currie, L., Patel, S., Cimino, J.: Heuristic evaluation of paper-based Web pages: A simplified inspection usability methodology. *Journal of Biomedical Informatics*, Volume 39, Issue 4, 2006.
- [2] Boehm, B., Basili, V.: Software Defect Reduction Top 10 List, *IEEE Computer*, Volume 34, Issue 1, 2001.
- [3] Dyba, T., Kampenes, V., Sjöberg, D.: A systematic review of statistical power in software engineering experiments. *Information and Software Technology*, Volume 48, Issue 8, 2006.
- [4] Fernandez, A., Abrahao, S., Insfran, E.: Towards to the validation of a usability evaluation method for model-driven Web development. *Proc. IV International Symposium on Empirical Software Engineering and Measurement*, USA, 2010, pp 54-57.
- [5] Fernandez, A., Insfran, E., Abrahao, S.: Usability evaluation methods for the Web: A systematic mapping study. *Information and Software Technology*, Volume 53, Issue 8, 2011.
- [6] Fons, J., Pelechano, V., Pastor, O., Valderas, P., Torres, V.: Applying the OOWS model-driven approach for developing Web applications: The internet movie database case study. In: Rossi, G., Schwabe, D., Olsina, L., Pastor, O.: *Web Engineering: Modeling and Implementing Web Applications*, Springer, 2008.
- [7] Kappel, G., Proll, B., Reich, S., Retschitzegger, W.: An Introduction to Web Engineering. In: Kappel, G., Proll, B., Reich, S., Retschitzegger, W.: *Web Engineering: The Discipline of Systematic Development of Web Applications*, John Wiley & Sons, 2006.
- [8] Laitenberger, O., Dreyer, H.: Evaluating the Usefulness and the Ease of Use of a Web-based Inspection Data Collection Tool. *Proc. 10th Software Metrics Symposium*, Bethesda - MD, 1998, pp. 122-132.
- [9] Murugesan, S. *Web Application Development: Challenges and the Role of Web Engineering*. In Rossi, G., Schwabe, D., Olsina, L., Pastor, O.: *Web Engineering: Modeling and Implementing Web Applications*, Springer, 2008.
- [10] Nielsen, J.: Finding usability problems through heuristic evaluation, *Proc. CHI'92*, UK, 1992, pp. 373-380.
- [11] Oinas-Kukkonen H.: Balancing the vendor and consumer requirements for electronic shopping systems. *Information Technology and Management*, Volume 1, Issue 1&2, 2000.
- [12] Oinas-Kukkonen, H.: *The Construct of Webflow for Capturing the Web User Experience*. 2011.
- [13] Ricca, F., Scanniello, G., Torchiano, M., Reggio, G., Astesiano, E.: On the effort of augmenting use cases with screen mockups: results from a preliminary empirical study. *Proc. 4th ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, Italy, 2010.
- [14] Rivero, L., Barreto, R., Conte, T.: Characterizing Usability Inspection Methods through the Analysis of a Systematic Mapping Study Extension. *Latin-american Center for Informatics Studies Electronic Journal*, Volume 16, Issue 1, 2013.
- [15] Rivero, L., Conte, T.: Improving Usability Inspection Technologies for Web Mockups through Empirical Studies. *25th International Conference on Software Engineering and Knowledge Engineering*, 2013.
- [16] Rivero, L., Conte, T.: Using an Empirical Study to Evaluate the Feasibility of a New Usability Inspection Technique for Paper Based Prototypes of Web Applications. *Proc. 26th Brazilian Symposium on Software Engineering*, Brazil, 2012, pp. 81-90.
- [17] Rivero, L., Conte, T.: Using the Results from a Systematic Mapping Extension to Define a Usability Inspection Method for Web Applications. *Proc. 24th International Conference on Software Engineering and Knowledge Engineering*, USA, 2012, pp. 582-587.
- [18] Shull, F., Carver, J., Travassos, G. H.: An empirical methodology for introducing software processes. *ACM SIGSOFT Software Engineering Notes*, Volume 26, Issue 5, 2001.
- [19] Signore, O.: A comprehensive model for Web sites quality. *Proc. 7th IEEE International Symposium on Web Site Evolution*, Hungary, 2005, pp. 30-36.
- [20] SPSS version 19.0.0 for Windows. Chicago, SPSS Inc, 2010.
- [21] Travassos, G., Shull, F., Fredericks, M., Basili, V.: Detecting defects in object-oriented designs: using reading techniques to increase software quality. *Proc. 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, USA, 1999, pp. 47-56.
- [22] Venkatesh, V., Davis, F.: A theoretical extension of the technology acceptance model: four longitudinal field studies. *Management science*, Volume 46, Issue 2, 2000.
- [23] Wohlin, C., Runeson, P., Host, M., Ohlsson, M., Regnell, B., Wessl, A.: *Experimentation in software engineering: an introduction (Book)*, Kluwer Academic Publishers, 2000.