Abstract. Usability evaluation is usually performed at the final phases of the software development process, when there is at least a system prototype – functional or not. Doing this evaluation earlier – at the design phase – can decrease the costs to repair eventual problems. For this kind of evaluation, artifacts as usage scenarios, interaction and task models from Human-Computer Interaction area are used. And as methods, formal and automatic usability evaluation ones are usually used. Considering that, this kind of artifacts and methods are not widely used, this paper focuses on the proposal of an informal evaluation method to be used with widely known diagrams: guideline review applied to UML diagrams. The method could be considered an easy and inexpensive alternative to be used by the development team during the design process of interactive systems.

Keywords: Human-computer interaction, usability, evaluation, model-based, diagrams, design.

1 Introduction

During the software development process, different kind of artifacts could be used to support the requirement analysis, the system planning and its construction.

Concerning these artifacts, white box models are normally used in the Software Engineering (SE) area, which prioritize how the system should be developed and not how it appears to the user [1]. The emphasis is on the internal structure of the system as opposed to the black box models which focus on the way the system will be presented to the user and which are used in Human-Computer Interaction (HCI) area.

When these models are used together, throughout the development process, they can ensure better understanding and documentation about the project to the development team. Considering issues related to the user and how he can reach his goals using the system, it is important to have an iterative process in which usability evaluations are made at different phases of the development process.

Thus, the sooner the problems are identified, the less the costs to repair them. The main goal of the work presented in this paper is “to predict the usability of a product or of some aspect of it” [2], bringing the evaluation to the early phases of the software development process.

To reach this goal, informal and empirical evaluation methods were investigated, looking for the best method to solve this problem. Informal – or inspection – methods are based on guidelines and/or the evaluator experience, such as heuristic evaluation
and guideline reviews. The empirical methods are those in which real users participate during the evaluation of the interactive system. They can vary from users’ interviews to tests with the users completing a task, for instance, usability and communicability tests [3].

We choose these methods because they are widely used by the HCI professionals and because they are manual methods. Manual methods depend on the human “eye”, considering subjective issues inherent to this kind of evaluation which is not usual using formal and automatic methods.

The method presented in this work proposes to anticipate usability evaluation of an interactive system, using UML diagrams, without modifications into the language, different from other proposals that suggest UML extensions or the adding of different diagrams to UML. We will present these proposals in the Related Works section.

2 Background

In this section we will introduce the diagrams used in the experiments and some usability evaluation methods.

2.1 Models

In this work, we studied Use Case Diagrams. This diagram is the most general and informal of the UML (Unified Modeling Language) – modeling language mainly used by RUP (Rational Unified process), generally used in the elicitation and analysis requirement stage. It presents a comprehensible language which gives the user a general idea of how the system works.

Along with the Use Case Diagram, Activity and Sequence Diagrams were used. According to [4], Activity and Sequence Diagrams present a more dynamic view of the system, while Use Case Diagrams present a more static and structural view.

The Activity Diagram aims to describe the steps to be performed to complete an activity, concentrating on the control flow of an activity while the Sequence Diagram represents a sequence (temporal order) in which the objects of an event exchange messages. A Sequence Diagram identifies the event that generates the modeled process and the actor that is responsible for this event, and also determines how the process must be performed and completed with method calls, fired by messages sent among the objects.

Therefore, in the studies presented here, we preferred the Use Case Diagram because it presents a more structural view of the system like the Task Models of HCI [3], while the Activity and Sequence Diagrams were preferred because they present a dynamic view of the system, where the user/actor interaction with the system modeled can be identified.
2.2 Methods

There are many known evaluation methods, and they can be classified in methods of formative or conclusive evaluation [5]. Within these methods, those more commonly used are the conclusive evaluation ones because the majority of them are easy to understand and to apply due to the fact that they are used when there is at least one prototype to be evaluated, which makes the evaluation more tangible. However, despite the fact that this type of evaluation is more widely used than the formative evaluation methods, this conclusive evaluation is only applied at the final stage of the system development and can increase the cost of repairs in the case of possible failures.

Informal Evaluation Methods. Heuristic evaluation is an informal evaluation method which has been developed as an alternative to deal with usability problems in little time and at a low cost [6]. Between three and five evaluators examine the interface individually, and, after that the evaluators meet to produce one single report that catalogs the problems encountered by each evaluator. Each problem is related to the infringed “heuristic” and its degree of severity. Guideline review is an informal evaluation method with a view to checking that the system is following the guidelines. The work of the evaluator is to agree or disagree with the checklist guidelines. In this kind of evaluation, contrary to heuristic evaluation, it is the quality of the checklist – in this case the guidelines – and nothing related to the evaluators’ experience, which determines the success of the evaluation.

Empirical Evaluation Methods. Within the empirical methods, the communicability tests [7] were used. The object of these tests is to evaluate the interface in relation to its communicability capacity – ability to transmit to the user, in an appropriate and efficient way, the intentions and principles for which it was designed. This method simulates the communication of the user with the design system, which is achieved through analysis and labeling of user actions, using a set of interjections. These interjections can be used when the user has expressed any difficulty in usage of the system. For example, when desiring a particular function which he/she cannot find. In this case the evaluator could use the expression “Where is it?”.

3 Proposed Method

According to the previous description, this work is within a wide project in which the necessity to anticipate interactive system evaluation for the design phase is discussed using HCI and SE models for evaluation.

The choice of these models is based on their popularity and acceptability by their communities (SE and HCI). This paper focuses on SE diagrams. This section discusses issues about usability evaluation based on these diagrams, e.g., Use Case and Activity Diagrams.
3.1 Phase 1: models and methods definition

In the first experiment performed in this project, we applied different HCI evaluation methods using both HCI and SE models with the aim of identifying which is the most efficient combination of informal method and model for discovering usability problems early in the design process.

In these studies, we used a UML modeling from a Digital Bookshop System [8], and HCI models were also constructed for the same system.

In this first experiment, we applied the following methods: heuristic evaluation [6], guideline review [9], communicability evaluation [7]. We applied these methods to Use Case Diagrams, Activity Diagrams, Sequence Diagrams, Goals Models [10], Task Models (HTA – Hierarchical Task Analysis) [3] and Interaction Models (MOLIC – Modeling Language for Interaction as Conversation) [10].

To choose the participants of the experiment, we defined the following profile: the participants had to have knowledge on UML modeling, HCI modeling and HCI evaluation. We selected twelve users and they were divided into different groups: four of them performed the heuristic evaluation, four carried out the guideline review and the remaining four participated as users in a model-based communicability test.

Concerning the guideline review, the larger number of problems found with this method were identified by every user that applied it. For instance, the problems “It doesn’t present options to Undo or Cancel” and “It doesn’t present Help and Documentation” were identified by every user that applied this evaluation method.

This resultant “pattern” reflects the objectivity of the guideline review method, in which the evaluator only needs to verify whether the system item is in agreement or not with the guidelines. If the checklist is more elaborated, it can cover a wide range of the interaction problems. However, this fact can restrict the achievement of results, which may interfere with the capturing of other problems.

The heuristic evaluation was the method that identified the larger number of problems. Most of the participants had difficulties to work with UML diagrams, mainly with the Sequence Diagram. According to the user, in the latter case, it was not the diagram structure that was not intuitive, but its content which was not dealing with the users’ needs.

Few problems were found with user tests. These can occur because of the difficulty to perform user testing on diagrams.

3.2 Phase 2: applying usability guidelines to models

Based on this data and the difficulties to carry out the tests with users working with models, we decided to only use inspection methods. We decided to define model-based HCI evaluation guidelines creating separate checklists for each of the four models, taking into consideration the particularities of each of them. In this experiment, we carried out an initial study to verify which guidelines would be applied to the models.

We used checklists related to each of the ten heuristics [6]. The checklists were based on checklists as [11], [12] and [13].
In this second experiment, we used the following models: a Use Case Diagram, an Activity Diagram, a Task Model and an Interaction Model. We decided not to work with the Sequence Diagram because of criticisms made by participants of the first experiment. The models used were the same as those used for the previous experiment – Digital Bookshop.

Sixteen participants contributed to this case study. The participants were undergraduates from the Computer Science course, who had a little experience in the diagrams (HCI and SE) and the evaluation methods used (HCI).

Many guidelines were not applicable to the models even though they had been changed. However, there were guidelines partially applicable and even guidelines directly applicable which could be adapted to be applied to models.

Because of these results, the participants’ suggestions and the previous identification of the need to adapt guidelines, an adaptation process was started, with the goal of making the guidelines applicable to the diagrams, always evaluating the usability of the system based on the diagrams and not analyzing the correctness of the model.

### 3.3 Verifying the proposed guidelines

After the guidelines elaboration, we performed a new experiment to verify the applicability of these guidelines to the models and to refine them.

Sixteen participants contributed to this case study, also undergraduate students, but not the same. In this experiment, we used the same system models (Digital Bookshop) and also the same previous UML diagrams: Use Case and Activity Diagrams. In general, there were no identified guidelines that were not applicable to the diagrams.

However, some guidelines still needed to be refined with the aim of improving their applicability. Therefore, we started a process of refinement of the guidelines (two of them are presented at the Table 1).

<table>
<thead>
<tr>
<th>Use Case Diagram</th>
<th>Activity Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>The options for the user must be organized in a logical way. (e.g., if there is obligatoriness between the tasks – use cases – it must use Include notation)</td>
<td>In phases where is necessary data input, the tasks must be described in a familiar terminology. (e.g., provide login and password)</td>
</tr>
</tbody>
</table>

### 4 Experiment

Aiming to validate the proposed method, we decided to verify the similarity of the issues identified on the system developed with the issues identified on the system modeling. We used Gym-Joe system, a web based system to manage a gym. This system provides the following features: maintain register of the users, maintain register of the employees, register of the suppliers, maintain register of the supplies and the forms of monitoring of the gym users.
We selected just some parts of the system to be evaluated in the experiment. The global goal of the experiment was to find out the correlation between of the application results, aiming to detect how many problems could be identified earlier, still in the modeling phase.

We used as a guide, the proposals of [14] e [15] as a complementary manner. The experimentation process used consists of the following steps:

- **Definition**: definition of the goals from the problem to be solved. This steps also consists the definition of the variables and the selection of the participants;
- **Planning**: experiment grounding. The conceptual preparation of the experimentation and the considerations about the reality, generality and experiment instrumentation;
- **Execution**: preparation, execution and validation of the data;
- **Analysis and Interpretation**: analysis and interpretation of the captured data of the experiment;
- **Packing**: documentation of the results and experiment organization, aiming to provide its replication (these experiment packing is still under development).

It is important to clarify that an experiment will never give a definitive answer for a question, but a specific answer for a given question. In the next subsections we will detail the steps to lead the experiment.

### 4.1 Definition

This phase we used a Goal Question Metric (GQM) [16] approach, it is a measurement to improve the software process guided by goals. The GQM approach starts from the goals definition – conceptual level – to establish the questions – operational level that try to characterize the process in terms of metrics – quantitative level.

- **Independent Variables** (process inputs): applying the guideline review on different artifacts (model and system).
- **Dependent Variables** (process outputs): correlation between the guidelines application results.
- **Subjects**: software developers (graduate students with experience on usability evaluation and interactive system modeling).
- **Object** (tool used to verify the cause-effect relation of a theory): an information system modeling designed and developed by students (toy example), in the context of a gym.
- **Parameters** (invariants): subjects experience on usability evaluation and on interactive system modeling.
- **Block Variables**: we applied a pre-questionnaire to capture the subjects’ profiles. This enables a balancing of the sample, turning the black variables to parameters.

To measure the response variables we need to have basic metrics that related could characterize the value of the response variables. We used the technique GQM [15] to identify these metrics.
Question: the results of the guidelines application to models are the same of the application to system?

Metrics: the method associated to the question is the correlation of the guidelines results in each approach. We defined the correlation, in this study, as the ratio between the number of the guidelines with same results and the number of guidelines with different results. We used SMC (Simple Matching Coefficient) [17] to calculate this correlation.

$$SMC = \frac{number\ of\ matching\ attribute\ values}{number\ of\ attributes} = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}}$$

Where:

- \(f_{00}\) = number of attributes where \(x\) is 0 and \(y\) is 0
- \(f_{01}\) = number of attributes where \(x\) is 0 and \(y\) is 1
- \(f_{10}\) = number of attributes where \(x\) is 1 and \(y\) is 0
- \(f_{11}\) = number of attributes where \(x\) is 1 and \(y\) is 1

Considering \(x\) the checklist applied on model and \(y\) the checklist applied on system. The result is 0 if the guideline is false (not satisfied) and 1 when the guideline is true (satisfied). SMC counts both presences as absences. Therefore, we can use SMC to find out correspondent guidelines (with same result) because they are binary, in other words, 0 (false) or 1 (true).

### 4.2 Planning

We choose a university environment to perform the experiment. Although some authors argue that is necessary to perform an experiment in a realistic environment [18], this kind of approach demands risks and costs that were not estimated in the scope of the presented work. In this context, we present:

- **Process:** we used an In-Vitro approach, in which a group of participants performed the experiment in a controlled environment. However, despite the environment is controlled, the participants were not grouped in the same environment at the same time. Each participant performed the experiment individually at different times, always observed by the researcher. This experiment is off-line, in other words, it was not performed during the industrial software development software process.
- **Participants:** the participants were graduate students of the Computer Science course. We did not use a probabilistic sample, but a convenience one.
- **Reality:** we studied a toy example problem and it is a system designed and developed by undergraduate students in a context of a gym.
- **Generality:** the experiment is specific and it is valid just in the scope of this study.
- **Instrumentation:** the goal of the instrumentation is to provide means to the leading and analysis of the experiment.
  - **Objects:** use case diagram, activity diagram, system developed, use case diagram checklist, activity diagram checklist.
4.3 Execution

This is the operational phase of the experiment, in which the participants are directly involved.

For instrumentation, every variables and resources were carefully established before the execution of the experiment. We introduced the experiment for each participant, presenting the goals, the technique, the motivation and the technical procedure to the experiment execution. Aiming to support the participants, we provided a tutorial telling how to apply the guidelines. We used a form to collect the data. We also took care with the ethical issues by the use of a consent form.

The participants captured the data using the forms, because the researcher cannot be in charge to the data resulting of the experiment. During the experiment, the researcher was available to the participants to answer some question.

4.4 Analysis and Interpretation

As the goal of the experiment was to verify the correlation between the guidelines results, we just applied SMC to the checklists answered by the participants. SMC between the use case diagram and the system was 0.19, in other words, 19% of the guidelines had the same result, whether false (0 – not satisfied) or true (1 – satisfied).

Between the activity diagram and the system, SMC is 0.58, in other words, 58% of the guidelines had the same result, whether false (0 – not satisfied) or true (1 – satisfied).

The use case diagram was the one that presented the minor correlation, it can have happened because their characteristic of informality.

On the view of participants, they were unanimous that the method is quick and easy application. Everyone said the use case diagram is too much informal and it does not have enough information to provide a system usability evaluation. They suggested that we use the use case specification.

5 Related Work

A great deal of attention has been focused on the usability of interactive systems, which corroborates with research about usability evaluation. Despite the knowledge of the importance to develop systems with quality of use, aspects related to the usability of the system are not the focus of our attention and are even less applied in bringing the usability evaluation to the design phase.

Meanwhile, there are proposals for adaptation of the RUP [19], aiming to direct more attention to the user interface of the system, adaptation of the UML [20] and to the addition of HCI diagrams into the UML diagrams set [21]. Concerning the model-
based usability evaluation, there are proposals of formal and automatic methods to evaluate usability from specific HCI models.

In [21], an integration of HCI Task Models – CTT (Concur Task Tree) [22] with the UML has been proposed with the aim of having a UML for interactive systems. The author discusses this new UML for interactive systems which could have the addition of the CTT into the UML set of diagrams, therefore creating a semantic mapping of the concepts of CTT to the UML meta model.

Still on the subject of the UML, in [20], an extension of the UML is proposed with the aim of representing the user interface, the UMLi. According to the authors, though the user interface represents an essential part of a system, the UML was developed with little attention to this fact. In this proposal, the tasks are modeled using an Activity Diagram extension – UML profile, instead of adding a totally new notation into the UML.

According to [23], in some cases the profiles focus on the modeling of graphic objects. However, this aspect begins to be resolved by IDEs (Integrated Development Environment) with the possibility of editing graphically the user interfaces. Still, according to the author, the question is not the edition of graphic aspects but the structure of the dialog between the users and the system. UML is established and both used and supported by a large community, which becomes relevant when exploring which is the best form to extract the maximum benefit from what the language provides as a basis for resolving problems that a software engineer may need to overcome when developing an interactive system.

In [19], RUPi is proposed. It is an adaptation of RUP, with the aim of integrating HCI aspects with its main workflows. According to [19], HCI concepts like human factors, guidelines, interfaces for all types and the model-based user interface production can contribute to the development of usable systems when applied to a software development process. This approach, however, also proposes the insertion of new artifacts during the development process. Not necessarily new diagrams but an addition of the better practices within RUP, as for example, new workflows.

Concerning a model-based usability evaluation, some authors like [24] and [25] defend the use of methods of automatic evaluation, which consist of verifying automatically and not manually the usability of a system, using formal methods. However, to perform this formal and automatic evaluation it is necessary that a formalization of the system and the user interface exists, which generates the necessity of other notations to represent the user interface. Besides this, an automatic evaluation may not be able to cover subjective aspects inherent in a usability evaluation.

As can be seen, the existing approaches aim to add new diagrams to UML or extend UML to the interaction modeling; or add new concepts into RUP. Model-based evaluation aims to use formal methods, generating an automatic evaluation, which induces high costs for modeling and also does not cover subjective aspects. The method presented in this paper proposes to anticipate usability evaluation of an interactive system, using UML diagrams, without UML extension or addition of new diagrams into the language.
6 Conclusion and Future Work

Taking into consideration the necessity to anticipate the usability evaluation with the aim of identifying usability problems within the time limit of the project and therefore reducing costs and finding solutions for possible problems, we proposed a model-based usability evaluation method.

The wide use of UML as a modeling language for the development of object-oriented systems and the limited knowledge of the development teams concerning HCI diagrams, lead us to believe that the proposed method should be used with UML diagrams.

Through the literature list made for this work various HCI methods, we can identify which are applied to models, of which the majority are automatic evaluation methods which fail to capture subjective data. With the aim of considering subjective issues in the model-based evaluation – like the meaning of words or the understanding of received feedbacks – this work proposes an informal evaluation method as opposed to automatic.

To achieve this, we needed to define which UML diagrams and which usability evaluation methods would be used and we carried out an initial case study, applying different evaluation methods on different diagrams. From this stage onwards, checklists for usability evaluation for the Use Case and Activity diagrams were defined. At the validation stage, with the aim of identifying the similarity between the identified problems, the checklists were applied for the system and for its UML diagrams.

Concerning the number of problems identified, if we analyze both checklists, twenty one problems were identified within the system and fourteen of them were identified using diagrams. In other words, 66% of the problems identified still in the modeling phase. It represents a reasonably high number of usability problems identified with UML diagrams, despite the fact that they are not developed to represent the user interaction with the system. It is important to highlight that this is a comparison between the same evaluation methods applied to two different stages of the software development. If other evaluation methods were applied to the system, it is probable that other usability problems would be identified.

Finally, this work can be considered as a “quick and dirty” evaluation method [9]. In this kind of evaluation, the designers have an informal feedback from the users or the evaluators to confirm if their ideas are according to the users’ requirements; emphasizing the quick contribution and not findings formally documented. The method herein proposed could present an alternative of low cost and quick application mainly to SE professionals with little experience in the HCI area, providing them a quick feedback of the system which they are developing.

This method may also be used by people with little knowledge in the field of HCI. The initial idea was to devise a method to identify problems in advance without the need of specialists to its application. However, during the experiment we observed, according to the evaluators, that there was a need for at least a little experience in usability, since the proposed guidelines were not as objective as expected, as it is necessary to interpret certain guidelines.

This informal feedback phase to the team is under development, we want to reach it in the next step, performing a case study which can be thought as “researches at the
real environment” according to [14]. In this next case study we will try to measure the effort spent using the method. Another alternative is to use the checklists as a guide to modeling interactive systems, because the designer could think of the user since the modeling phase. As some participants commented that they do not use UML, we are working on evaluation of different artifacts such as scenarios and user stories.

References


