Journal of Universal Computer Science, vol. 19, no. 8 (2013), 1046-1065
submitted: 30/11/12, accepted: 29/3/13, appeared: 28/4/13 © *J.UCS*

Applying Usability Engineering in InterMod Agile Development Methodology. A Case Study in a Mobile Application

Begoña Losada, Maite Urretavizcaya, Juan-Miguel López-Gil
Isabel Fernández-Castro
(University of the Basque Country UPV/EHU, Donostia-San Sebastian, Spain
{b.losada, maite.urretavizcaya, juanmiguel.lopez, isabel.fernandez}@ehu.es)

Abstract: This paper explains when and how to integrate aspects of usability engineering in the agile development process proposed by the InterMod methodology. The aim of InterMod is to facilitate the accurate development of high-quality interactive software. This is accomplished by means of agile software engineering activities and continuous assessment in which certain usability evaluation techniques have been suitably integrated. Such integration brings benefits to the early steps of the development process. On the one hand, this integration promotes development tailored to users' expectations. On the other hand, it helps to plan the agile process of activities. In terms of developing a mobile application, the use of prototypes in early stages of the development process has been considered as being quite beneficial for organizing the workflow. Using heuristic evaluations and user tests has also been valuable when grouping evaluations in specific iterations.

Keywords: Agile development methodologies, software development, usability engineering
Categories: D.2.10

1 Introduction

Interactive systems involve various aspects that are related to issues in Human-Computer Interaction (HCI) that demand improvements in standard software life cycles. These systems must be observed and evaluated in order to determine how users interact with them so more usable systems can be developed [Dix 03]. User Centred Design (UCD) approaches the development of interactive products by involving users throughout the design and development processes, seeking to optimize the system's usability [Stone 05]. In particular, classical usability engineering methods used in UCD [Mayhew 99] [Nielsen 93] introduce explicit usability goals into the development process, and gather requirements in a major up-front specification effort. These methods state that evaluation is an essential activity that must begin as early as possible in the development process [Granollers 05].

Agile software development (AD) arose from the need to discover better ways of developing software [Beck 01]. AD is an alternative to traditional software development approaches that emerged after numerous project failures. A failure in a project may not only result in its cancellation or late delivery, but it may also mean that the final product does not meet expectations [Larman 03] and thus it is never used. AD processes quickly adapt to project changes and promote incremental software delivery in short time periods. Therefore, they continuously analyse the evolution in requirements instead of recording an overall requirement collection prior to implementation. Agile software development methodologies share some crucial features with usability engineering, such as iterative processes, a human-centred focus, continuous testing, and a commitment to quality in the developed software. Nevertheless, there are also notable differences, such as the up-front design, the continuous changes throughout the process and documentation. Although there have been some proposals to combine both approaches [Hellman 10] [Hussain 09], it is still difficult to address these concerns.

This article is focused on incorporating usability engineering into the agile software development methodology called InterMod. InterMod manages software development by means of User Objectives (UO), i.e. user desires that can be met by one or more functional and/or non-functional requirements. Throughout the different iterations, the development teams work in parallel and perform activities to obtain developmental models related to different UOs. Every UO is a complete logical unit and at the same time a part of the final result. All these InterMod processes, together with the use of a specific requirements model, allows the HCI and AD disciplines to be integrated in an optimal way.

We used and validated this integrated approach during the development of a mobile application. The usability techniques used had to be consistent with the model to be evaluated, and the development team determined the usability techniques to be used throughout the whole process. The general conclusions concern (1) the integration process itself, which includes grouping evaluations, the need for involving HCI experts in development teams, and the balance between agile development and evaluation, and (2) the application of the integration process when building mobile environments, whose particularities must be taken into account during evaluation processes. Thus, we have demonstrated that integrating usability evaluation techniques produces verifiable improvements in the agile development process.

The rest of the paper is structured as follows. Section 2 describes related work, and section 3 details when and which usability engineering techniques integrate within the InterMod agile methodology. In section 4, the specific usability evaluation techniques applied in the development of a mobile application called FindMyPlace are presented. Section 5 discusses the agile and usability aspects, and section 6 outlines our conclusions.

2 Related Work

Now that agile methodologies are growing in popularity in the software development industry, a major effort is being made to integrate them with HCI. To date, modelling languages that formalize Software Engineering, such as Unified Modelling Language (UML), are being used to perform model-driven design. In addition, new proposals extend UML in order to model aspects related to usability. Some authors have extended UML to include task models [Nunes 00], while other authors have sought equivalences among UML's diagrams and HCI's models. For instance, they relate UML's Use-case and Activity diagrams with HCI's User-task models [Markopoulus 00]. It is also usual to improve UML activity diagrams so they represent HCI concepts such as task order or type [Pinheiro 02]. However, although it correctly specifies any applications' structure and functionalities, UML is not commonly used to specify usability-related information [Sutcliffe 05]. This may be due to the fact that these new UML extensions have produced more formalism and less understanding [Löwgren, Stolterman 07]. [Memmel 07] stated that "formal models used in Software Engineering and excessive documentation derived from HCI such as style guides are far too expensive for the design of interactive products in any case".

Despite the difficulties of integrating AD and HCI, numerous efforts are being made to do so. They are motivated by the fact that software with poor usability can reduce productivity and user acceptance [Hellman 10]. In fact, most modern approaches in both disciplines are not opposed to collaboration. Discussion about agile approaches to user interface design has resulted in a movement within the HCI community that has begun to reconsider heavy-weight lifecycles [Constantine 02b]. This has led to the development of light-weight approaches, such as eXtreme Usability [Federoff 08], Agile User-Centered Design [Sy 07] and Agile Human-Centered Software Engineering [Memmel 07]. Constantine also proposed a light-weight approach called Usage-Centered Engineering [Constantine, Lockwood 02a], which described navigation architecture and the interaction design scheme on the basis of fast and comprehensive task modelling.

Several solutions have been suggested to arrive at proper integration. Some of them, such as [Constantine, Lockwood 02a], propose to use models such as the ones described by Ambler's agile modelling [Ambler 13]. Others present an integrator model to bridge the gap between AD and HCI [Blomkvist 05] [Losada 12a]. Most of them recommend integrating several HCI techniques in their agile processes, such as field studies, the Personas technique, usability tests and usability evaluations with experts. In this sense, some informal methods typically used in agile development, such as user stories or prototypes, are close to HCI practices and are used as a link between both disciplines [Ratcliffe 11]. Some authors propose the use of scenarios, a usability engineering technique that can be equated with the user stories employed in AD [Lee 06] [Obendorf 08]. [Hellman 10] and [Kane 03] propose using discount usability techniques, which were originally proposed by [Nielsen 93] to perform usability evaluations with a limited scope and cheap evaluation methods. Heuristic evaluations [Nielsen, Molich 90] and Wizard of Oz testing [Nielsen 93] are some examples of discount usability techniques.

The attempts to integrate usability in agile approaches can be divided into two groups: (1) proposals that explore ways of incorporating usability into the most popular agile methodologies (such as Scrum or XP), or (2) proposals that specify new usable and agile processes. Most proposals found in the literature [Hellman 10] [Hussain 09] belong to the first group. Other proposals, such as the ones that integrate HCI with specific development approaches, as is the case with Test-Driven Development, can be also included in this first group [Robles 11]. The proposal by [Sy 07] and the one presented in this work can be included in the second group. A mixed proposal is eXtreme Scenario-based Design [Lee 09], which uses the Central Design Record in a XP model to support synchronization activities, help the usability engineer plan, and run usability evaluations. Additionally, Extreme Usability combines XP and Usability Engineering, and U-SCRUM [Singh 08]), a variant of the Scrum methodology, proposes to consider two product owners, one focused on usability and the other on more conventional functions.

3 Integrating Usability Engineering in InterMod

InterMod is an agile methodology that aims to help with the accurate development of high-quality interactive software [Losada 12a]. Agile methods differ from predictive software development processes since they provide a dynamic adaptation to the new contexts that arise during project execution. Thus, they emphasize continuously improving and adding functionality throughout the project life [Beck 01]. However, because agile methods require pieces of working software to be continuously delivered, the organized development of the interface may be ignored. To solve such issues, InterMod proposes to apply the Model Driven Architecture approach [OMG 03] in order to develop interactive software based on the models generated and evaluated during a project's progress according to user objectives. A User Objective (UO henceforth) is a user desire (e.g., "buying a custom t-shirt securely" or "reserving a meeting room at the workplace using my mobile phone") that can be attained by one or more functional and/or non-functional requirements.

Like agile methodologies, InterMod proposes to organize a project as a series of iterations, and to distribute the work among the iterations according to different *activities* of the UOs. In this way, the project development process can be carried out in parallel by different workgroups. The activities in the iteration can be Developmental Activities or Integration Activities. Three Developmental Activities (DAs) are associated with each UO: Analysis and Navigation Design (DA-1), Interface Building (DA-2) and Business-Logic Coding (DA-3). Furthermore, three Integration Activities (IAs) assure the correct incremental progress of the project: Requirement Models Integration (IA-1), Interface Integration (IA-2) and Code Integration & Refactoring (IA-3).

All developmental and integration activities are driven by models. To begin, the User and System models collect the characteristics that influence the entire project and that can be refined throughout the iterations. The User Model configures aspects such as colour preferences, font, size, certain limitations such as colour blindness, deafness, vision loss, etc. The System Model helps to define characteristics of the platform type (e.g., device type, window size, colour, etc.), but it can also collect other aspects related to the application, like security, logo, etc. These two models are evaluated from the beginning and throughout the project. It is necessary to ensure coherence with both the system characteristics and the users' needs and preferences throughout the entire application. The developmental models may be created in parallel for different UOs and/or different activities and they must be continuously evaluated (Fig. 1, left side). In particular, DA-1 and IA-1 activities deal with the Requirements Model (M-1), which in InterMod is called the Semantically Enriched Human-Computer Interaction (SE-HCI) model (see [subsection 3.1]). DA-2 creates the Presentation Model (M-2) for some of the UOs that have been previously designed and evaluated, and IA-2 integrates the M-2 of some UOs. The M-2 of a specific UO establishes the graphical elements gathered from the M-1 abstract model. Finally, DA-3 and IA-3 deal with the Functionality Model (M-3), which guides the implementation in a particular programming language. For a concrete UO (UOi), M-3 inherits the behaviour characteristics from the M-1 of this UOi (Fig. 1, right side).

InterMod (Fig. 1, left side) defines an initial step, step 0-"Analyse the Overall Project", and continues with a sequence of iterations. At Step 0, an initial analysis of the whole project determines: (a) the starting UOs, and (b) the general design decisions. The starting UOs will be the ones that are most important or needed by the end users. Additionally, it is important to draw up the System and User Models. For the sequence of iterations, each iteration involves three steps: Step 1.i -"Build the User Objectives List", Step 2.i-"Plan Parallel Iteration", and Step 3.i-"Perform Iteration Activities", where i is the i-th iteration. These steps make it possible to coherently fragment the project into UOs, and the UOs into activities. The UOs List is updated at Step 1.i with the new UOs derived either from the new needs for the project (Direct UOs) or from previous developments (Indirect UOs). At Step 2.i the UOs to be developed are determined, along with the activities to be carried out for those UOs, and how the chosen activities will be distributed to the workgroups (if there is more than one). Finally, at Step 3.i of the iteration, each workgroup performs its activities as established in the plan.

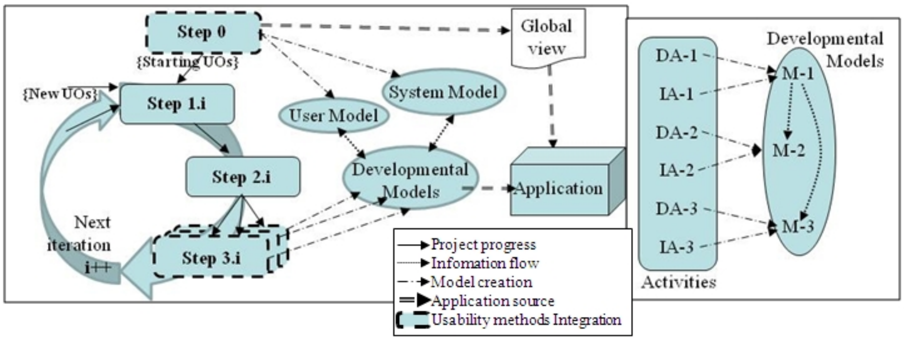


Figure 1. InterMod methodology (left), and the relations between activities and models (right)

Evaluation techniques can be applied from the early stages of the system development process [Mayhew 99], and we propose to start using them in Step 0. Usability evaluations are particularly relevant for Direct UOs as they reflect direct user needs, so a group of end users must necessarily be involved in evaluating them. However, in order to expedite the project, Indirect UOs may be evaluated by usability experts only. Inspection, inquiry and test methods [Nielsen, Mack 94] can be combined to obtain a more comprehensive result when analysing the usability of the models created in Step 3.i. Other HCI evaluation methods that rely on measures of user satisfaction and task efficacy and efficiency are also proposed. [Section 3.2] includes the techniques that can be integrated in the methodology.

3.1 The SE-HCI model supports the project requirements

The project requirements are defined by the SE-HCI Model [Losada 09], which incorporates the User Task Model (Fig. 2) and is influenced by the *User* and *System Models*. The User Task Model is a classic element in Model-Based User Interface Development [Paternò 99] [Limbourg 05] and can be used to describe the user's performance in completing each UO. The SE-HCI Model also describes: (1) the interactions that can be suitably established between the users and system at the user interface level [Paternò 99] and their possible temporal relationships; (2) the incorrect uses of the application that can occur through possible interactions over the interface navigation, and thus they define, together with the correct uses obtained from the Behaviour Model of the User Task Model, the semantics of the application through the *SE-HCI Communication & Behaviour Model*; and (3) the basic characteristics—gathered in the *SE-HCI Prototype Model* (colours, sections, button types, etc.)—that draw the basic appearance of the application.

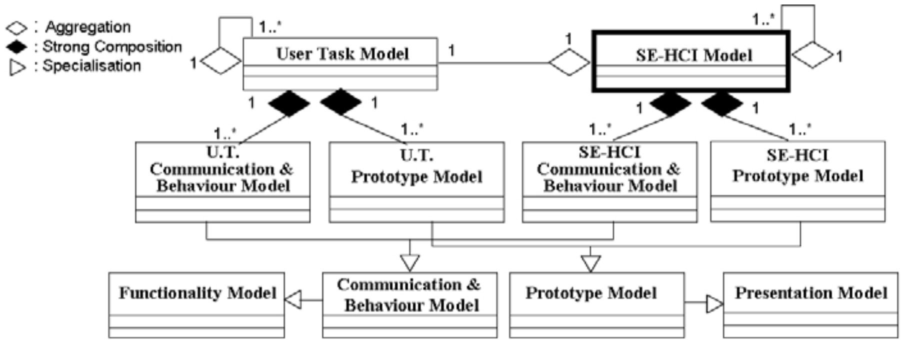


Figure 2. Hierarchy of models related within the SE-HCI Model

Nielsen defines the discipline of usability engineering as a software engineering process that includes usability considerations in software development process [Nielsen 93]. One of the important features of usability engineering is the inclusion of a usability specification. Therefore, the specification of requirements must also focus on the features of the user-system interaction that contribute to product usability [Dix 03]. In this sense, the SE-HCI Model is devoted to capturing these kinds of interactions in a usability specification.

3.2 Usability techniques applied in InterMod

The collaboration of end users is fundamental in order to determine the starting UOs. UOs are gathered from information obtained through usability engineering methods. The selection criteria are based on the priorities of the features that are needed. Inquiry methods are the most suited to gathering information about user desires and needs, which would be expressed by the starting UOs. Questionnaires, unstructured and structured interviews, focus groups and field observation [Nielsen, Mack 94] are some of the most common techniques that can be applied. Questionnaires and interview-based protocols are especially useful for gathering data by querying users about the global view of the system with regard to various aspects, such as navigation, presentation and behaviour. Additionally, the System and User Models are collected and validated by means of inquiry methods applied to end users in the presence of usability and domain experts.

Each developmental and integration activity creates one or more models that must be evaluated with the techniques that are best suited to the models' characteristics. InterMod advocates the use of test-based methods as the best way to evaluate the M-1 model. The evaluation of M-1 captures errors related to requirements, and it can also discover and anticipate interface errors (*SE-HCI Prototype Model*) and even business logic errors (*SE-HCI Communication & Behaviour Model*). The SE-HCI Model has got enough information to automatically generate low-fidelity prototypes. From this point, designers, users and developers can jointly carry out the evaluation. Typically, representative users perform specific tasks on the system (or prototype), and evaluators observe and collect the interaction data through thinking aloud protocols [Lewis 82], performance measurement [Dumas, Redish 93] and coaching methods [Nielsen 93]. Inspection methods involving usability experts and/or end users are proposed for evaluating the M-2 models. The results of heuristic evaluation [Nielsen, Molich 90] or cognitive walkthrough [Wharton 94] provide a good source of feedback that serves to improve specific elements of the user interface. Finally, the M-3 models may be evaluated with test-based and inspection methods. Moreover, they should be checked by using software quality specifications, such as [ISO/IEC 9126-1 01].

4 Usability Techniques Applied in FindMyPlace

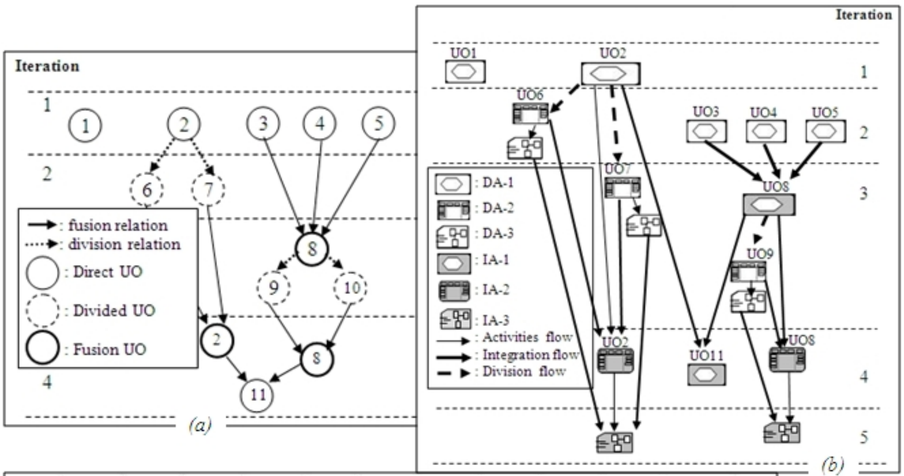
In this section we first describe an abstract of the FindMyPlace developments and some views of the project's progress step by step. Then, a detailed vision of how usability evaluations were performed during the agile development process is presented. In particular, we describe the usability techniques used in the following InterMod milestones: at the beginning (Step0) to capture user needs and the general design decision of the application, and throughout the iteration to evaluate the created models (Step 3.i): M-1.Requirements Model, M-2.Presentation Model, and M-3.Functionality Model.

4.1 Development of FindMyPlace

FindMyPlace is a mobile application devoted to helping users to find physical locations in closed scenarios. It allows different areas to be located inside a building, such as classrooms, laboratories, faculty offices, seminar rooms, etc. The application uses the building plans in the computer science faculty and maps of its surrounding area. GPS and indoor triangulation technology are also used to achieve this aim.

Regarding mobile applications, [Nielsen 11] performed a series of usability studies that stated that mobile users face severe usability problems in attempting to get things done on websites, whether through dedicated mobile sites or traditional desktop-optimized sites that are rendered through a mobile browser. Indeed, such evaluation approaches may not be directly applicable to mobile settings. Thus, it is not trivial for evaluation methods to fulfil the need to properly integrate real-world settings or simulated context during the evaluation process. A relatively big controversy has arisen around the goodness and relevance of results in mobile usability tests when mobility and contextual factors are considered. Several studies were performed to contrast laboratory studies with field tests, and their conclusions suggest that, in most cases, the most relevant feedback is obtained from laboratory studies [Kaikkonen 05]. Other authors concluded that some HCI practices on their own are not enough for mobile phones due to their small size and their limited user input capabilities [Kangas 05]. However, it should be noted that there has been great evolution in these devices, which is apparent on bigger screen sizes, and a significant increase in interaction capabilities.

Therefore, in this paper we attempt to demonstrate the feasibility of HCI practice with a case study in a mobile application that is evaluated step by step with usability techniques integrated in an agile methodology. The iteration-by-iteration development of FindMyPlace [Losada 12b] is summarized in Fig. 3, while the evaluation processes are detailed in the following subsections.



	UO1	UO2	UO3	UO4	UO5	*UO6*	*UO7*	UO8	*UO9*	*UO10*	UO11
M-1	X2	X1	X2	X2	X2	D2	D2	I3	D3	D3	X4
M-2		I4	F4	F4	F4	X2	X3	I4	X3	F4	
M-3		I5	F5	F5	F5	X2	X3	I5	X3	F5	

(c)

Figure 3. Progress of FindMyPlace in the iterations: (a) by UOs created (Step 1.i), (b) by Activities planned (Step 2.i) and (c) by Models created and evaluated (Step 3.i).

Fig. 3.a depicts the progress of the project by taking into account the creation of the different UOs: Direct UOs and Indirect (Divided and Fused) UOs. Fig. 3.b illustrates the project's progress by showing the activities that were planned and carried out. The table in Fig. 3.c summarises the models that were developed and evaluated. Each cell of the table is fed with a code composed of a letter (X, I, D or F) and a number, e.g. X1 or I3. Each possible code indicates how and when the model was created. The letter expresses how the model is obtained, i.e. X: developmental activity (DA), I: integration activity (IA), and D/F: UO division/fusion process. The number identifies the iteration in which the model was created and satisfactorily evaluated. The white cells contain information about Direct UOs and the shaded ones correspond to Indirect UOs. Italics are used for UOs obtained by division, and the lack of italics indicates fusion.

4.2 Initial analysis of the project through usability techniques (Step 0)

At Step 0, a user session was prepared in order to identify the main leading UOs of the application by means of inquiry methods. This process is fully explained below.

Objectives: To obtain information about the end users, their needs and priorities. This information would be the basis for defining the initial list of UOs and thus beginning the development process of the FindMyPlace application.

Participants: 20 volunteers participated in the user session. All of them were students in the first year of the Computer Science degree programme at the University of the Basque Country. They were also selected as possible end users of the application. In addition, all the project developers attended the meeting in order to answer the users' questions.

Instruments: The questionnaire was developed as a GoogleDocs form. It was divided into three different parts: (A) User demographic characteristics: Internet connection on the mobile phone, use of a Smartphone or a basic phone, use of GPS on the phone, etc.; (B) an open questionnaire collected, via a text box, all possible features that users would like the intended application to have; (C) a structured questionnaire about the 13 possible user needs proposed by the project developers. By using Likert scales users had to evaluate each need by indicating their degree of agreement and priority.

Procedure: All the students were summoned to a meeting in order to complete the questionnaire. First, they received a brief explanation about the purpose of FindMyPlace, and then the overall session and the different parts of the questionnaire were also described. Finally, they were encouraged to ask questions about completing the questionnaire, which would be answered by the project's developers, who were present in the laboratory. The whole process lasted for approximately 20 minutes.

Results: The information gathered let us classify the end users and identify their functional and non-functional desires regarding the application's behaviour. Eighty per cent of them had mobile phones with Internet connection, 70% had a Smartphone and 45% routinely or occasionally used maps on their mobile phones. Sixty per cent used the Android platform, which confirmed its selection as the application's platform. The functional desires identified gave rise to a list with five initial UOs. All UOs had the same priority at the beginning of the development process. The UOs were numbered consecutively and defined with a descriptive name: *UO1-Guiding to a given location, UO2-Showing distribution of spaces in the building plan at different levels of detail, UO3-Locating a teacher's office in the building plan, UO4-Locating a laboratory in the building plan* and *UO5-Locating a special location in the building plan*. In addition, the global view of the application was based on three search options: "People", "Building Plans" and "GPS Guide".

4.3 Evaluation of the Requirements Model (M-1)

The DA-1 activities developed for UO1 and UO2 generated the M-1.Requirements models (iteration 1). These models were evaluated over prototypes generated by the Diagram Tool [Losada 09]. Diagram allows the creation and graphical visualization of the SE-HCI model in a hierarchical representation by means of different kinds of user and system tasks. The automatically generated prototypes simulate certain system features, such as navigation, behaviour and presentation. The evaluators performed the tasks included in the SE-HCI models of both UOs, and expressed their opinions according to the Thinking Aloud technique [Lewis 82]. Several problematic aspects related to the terminology used in the interface were identified, especially for UO1. After correcting these problems, a complementary evaluation involving end users and more realistic prototypes for UO1 was proposed. However, UO2 was ready for further development.

In iteration 2, we applied different usability techniques to evaluate M-1 for UOs {1, 3, 4, 5}. Paper prototypes [Mayhew 99] developed from the previous SE-HCI navigation diagrams, the Thinking Aloud technique [Lewis 82] and observation and interviews [Nielsen, Mack 94] were used. The user session was carried out as follows:

Objectives: To detect problems in the user interface related to navigation and interaction before developing any code.

Participants: 18 volunteers participated (of the initial 20) in the session and 3 usability experts were present to guide the session. They also acted as evaluators.

Instruments: The experts developed a paper prototype composed of 16 screens that represented the user interfaces for all UOs under evaluation. The screens included the characteristics established in the SE-HCI model concerning navigation, behaviour and other user interface aspects. Fig. 4.a shows some of the screens and a navigation sequence that is dependent on the actions performed on the interface.

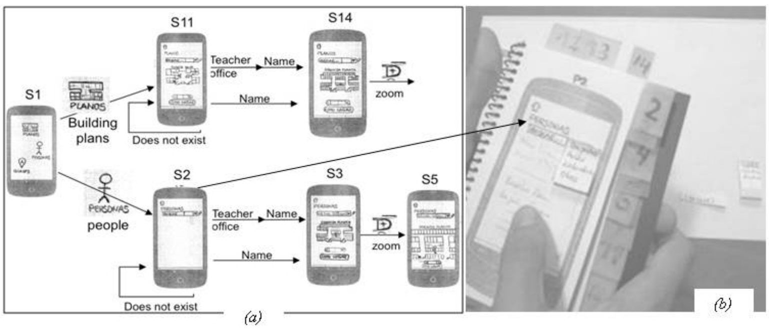


Figure 4. UO3 evaluation: (a) sequence of screens and (b) the S2 paper prototype

The screens simulated the user interface of an Android mobile phone and were developed on a paper sheet template. All 16 screens were bound as a booklet, and a series of post-its were used to properly simulate the navigation sequence through the prototype (Fig. 4.b.). Three booklets were prepared, one for each evaluator.

Procedure: An evaluator near each user presented the prototype booklet and undertook a simulator function. In this way, he/she could replicate the system response for each user action by adding post-its to the proper screen, and then by navigating through the screen prototype.

The evaluators presented a series of User Objective Scenarios to be performed by means of the paper prototype. The User Objective Scenario concept derives from the Task Scenarios concept [Mayhew 99] when this is applied to UOs. An UO scenario is a hypothetical story designed by the tester to help the end user to evaluate a UO through a given situation. Tab. 1 shows each UO Scenario; X and Y represent the hypothetical names of a teacher and a laboratory, respectively.

All users performed all the UO Scenarios following a randomly generated sequence. In this way possible correlation effects derived from the execution order were avoided. In addition, the evaluators annotated all the users' actions. After completing the scenarios (Tab. 1), a brief interview was conducted to capture users' opinions about the system and aspects that could be improved or included. The session lasted for approximately 30 min/user.

UO	User Objective Scenarios	Achievement
1	I am in the hall and I want you to guide me to teacher X's office	16.67%
3	Show me teacher X's office in the building plan	88.89%
4	Show me where lab Y is located in the building plan	94.44%
5	Show me where the cafeteria is located in the building plan	100%

Table 1. User Objective Scenarios and task completion percentage

Results: Tab. 1 includes the completion percentage of the UO scenarios. From the data, it is clear that UO1 requires a major iterative redesigning process, as most users were not able to complete it. The possibility of determining this fact at such an early stage of development is a clear advantage that the SE-HCI model provides. However, the UOs related to searching for specific targets in the building plans were completely finished nearly every time, although minor changes were still needed. Apart from efficacy-related measures, evaluators also collected data to assess efficiency (see Tab. 2). It must be noted that these data provide information about efficiency only for those scenarios that were successfully completed by users and not for the others.

	UO3	UO4	UO5
Non-optimal paths	6.25	11.76	16.67
Click on incorrect elements inside the correct path	0	1.33	1.00
Incorrect data input	0	1.67	3.00

Table 2. Efficiency measures for the objective scenarios

"Non-optimal paths" expresses the percentage of users that followed a non-optimal path to fulfil a UO Scenario. The number of clicks that users made on the interface elements belonging to the correct path but that did not lead to the correct fulfilment of the objective scenario was also noted. Finally, the number of times that users included incorrect data in a textbox was also registered.

Efficiency measures provide valuable information regarding the use of the interface and the aspects that need improvement. For instance, UO5 was the objective scenario where users introduced more data incorrectly (Tab. 2), despite being the most effective one (Tab. 1). Thus, in this case, the efficiency measures indicate the need to improve the user interface, but they also prove that it is good enough for users to rewrite the input text adequately when a mistake occurs. We have also applied a t-test to the rate of UO scenarios completed, non-optimal paths followed and incorrect clicks made, for the three groups of users. In all cases the $p\text{-value}>0.20$. When applying a t-test to incorrect data, the $p\text{-value}>0.08$. These facts are remarkable, as they indicate that there are no statistically significant differences in user performances when considering the mobile platform, the type of phone and the use of maps.

4.4 Evaluation of the Presentation Model (M-2)

In iteration 4, a heuristic evaluation was used to determine the suitability of the M-2 of UO2. The evaluation process was as follows.

Objectives: To detect navigation and interaction problems in the user interface early in the development process.

Participants: Three usability experts performed the evaluation.

Instruments: A functional version of the application that implemented UO2 was installed on a Samsung Galaxy S mobile phone platform. Every evaluator used a mobile phone to evaluate the user interface according to Bertini's eight heuristic principles, which were divided in 35 sub-heuristics [Bertini 08]. The sub-heuristics were assessed according to the Severity Ranking Scale (SRS) [Nielsen, Mack 94], which weighed usability errors according to their importance. The sub-heuristics were also analysed to determine whether they were applicable or not applicable (NA).

Procedure: As a first step, evaluators studied the interface and then checked all the heuristics and their corresponding sub-heuristics. Every applicable sub-heuristic was ranked by the experts according to Nielsen's SRS, ranging from 0-4 (*no usability problem at all* to *usability catastrophe*). Once the individual evaluation processes were finished, the evaluators met and presented their findings and agreed on a list.

Results: Tab. 3 shows the results. The focus of the evaluation was on identifying potential usability errors in applicable sub-heuristics, although NA sub-heuristics were also represented. The results did not show any usability catastrophes. Forty-three per cent of the sub-heuristics did not present any usability problems, and over a quarter of the sub-heuristics were NA. The two major usability problems found were related to the limited number of ways that the application could be manipulated, as manipulating it with only one hand—a highly probable scenario in a mobile context—is a difficult motor skill for the user. Some minor usability problems were detected: (1) the screen is small when the landscape mode is active; (2) images are not displayed like a photo gallery, which lacks continuity; (3) the button distribution on the initial screen in landscape mode is inadequate; and (4) there is a lack of personalization options in the application. Finally, there were still some superficial problems concerning reading and navigating the screen contents, which requires: (1) providing greater flexibility when initializing the user interface elements; (2) improving the icon aesthetics when they are selected and displayed; and (3) clarifying two error messages related to visualizing the building plans.

SRS	**NA**	**0**	**1**	**2**	**3**	**4**
Number of sub-heuristics	9	15	4	5	2	0
Percentage	0.26	0.43	0.11	0.14	0.06	0.00

Table 3. Results of the heuristic evaluation on the prototype

For future developments, some sub-heuristics primarily related to privacy and security issues will be considered. In the current development, the building plans were located within the application itself, so there were neither privacy nor security concerns about how the information was obtained. These kinds of issues will likely appear when data is transmitted from a server to the application, even though the provided information is publicly accessible at the university's website.

4.5 Evaluation of the M-3.Functionality Model

The M-3 of UO2 was evaluated in iteration 5 by means of usability tests, including user satisfaction and efficacy and efficiency measures.

Participants: At the time this evaluation was carried out, it was a new academic year and so it was necessary to recruit a new group of end users. The group consisted of 12 students in their first year of college whose adeptness with the user model was checked. All of the new end users owned smartphones, 91% of which were mobile phones with Internet connection, 66% of which had the Android platform and 83% of users routinely or occasionally maps used on their mobile phones.

Material and Instruments: Again, Samsung Galaxy S mobile phones were used. They had a running application with all the functionalities related to UO2, and a module to gather usage logs during user interaction based on the Android-Logging-Log4J library. The demographic questionnaire used in Step 0 was used to check the adeptness of the new user group. At the end of testing session, users had to complete the CSUQ satisfaction questionnaire [Lewis 01]. This includes 19 statements that users rank from 0 to 9 depending on their degree of agreement. Users could also mark a statement as being not applicable for the task, and make any other comment about it.

Procedure: First, end users filled in the demographic questionnaire to check that they fit the user model. Then, they carried out four tasks in the application with no time limit. Like in the evaluations of M-1 in iteration 2, the evaluators presented a series of User Objective Scenarios to be performed, also in a randomly selected order (see Tab. 4). After completing all tasks, participants filled in the satisfaction questionnaire, and then there was a short interview with users so they could express their opinion about the application. The session lasted for approximately 15 min./user.

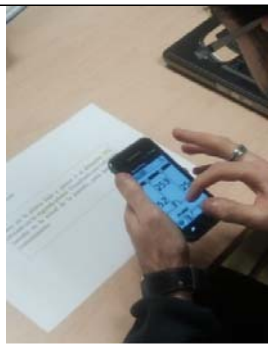
<table>
<tr><th>Task</th><th colspan="2">UO2 Evaluation Scenarios</th></tr>
<tr><td>1</td><td>I am on the ground floor and want to go to the cafeteria, located on the ground floor. Visualize it centred in the screen, with a suitable size as to be properly displayed</td><td rowspan="4"></td></tr>
<tr><td>2</td><td>I am on the ground floor and want to go to laboratory E08, located on the mezzanine. Visualize it centred in the screen, with …</td></tr>
<tr><td>3</td><td>I am on the ground floor and want to go to the school secretary, located on the first floor. Visualize it centred in the screen, with …</td></tr>
<tr><td>4</td><td>I am on the ground floor and want to go to office 253, located on the second floor. Visualize it centred in the screen, with …</td></tr>
</table>

Table 4. UO2 Evaluation Scenarios. Snapshot of a user handling the mobile phone while evaluating task 4

Study process and Results: The log file obtained by the application requires a pre-process to facilitate its subsequent study (Fig. 5.a). There are two main kinds of user actions over the interface: to press some buttons, and to touch the building plans to move them or zoom in. Thus, only some of the information collected in the log file is required, i.e. time, button pressed (e.g. "building plans") and identification of the action on the screen expressed by a number code (e.g. 0, 261, 2, etc.).

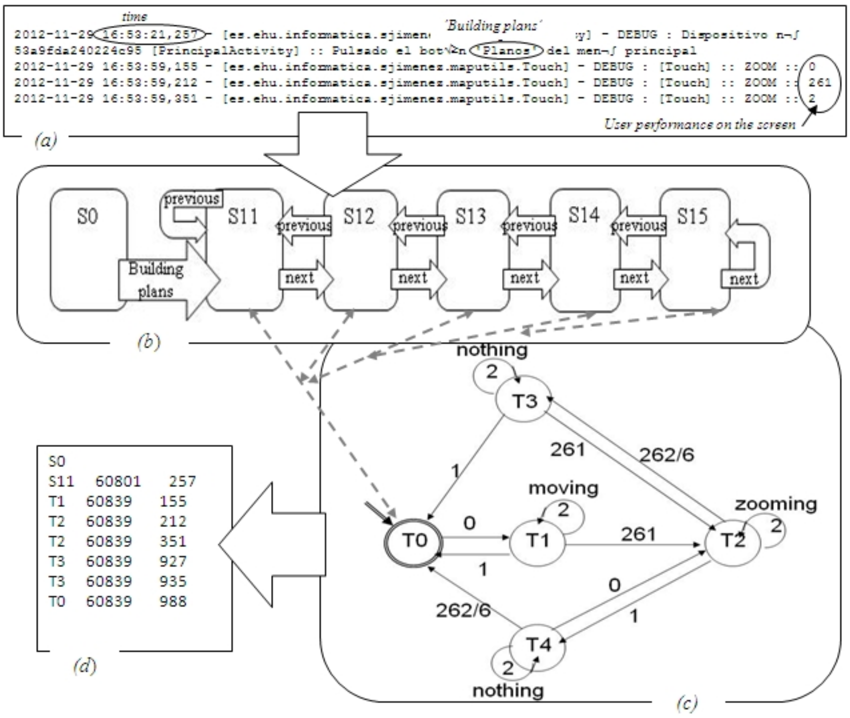


Figure 5. The log file pre-process: (a) Initial log file fragment, (b) UO2 screen sequence generated by pressed buttons, (c) FSM of touch performance, and (d) New log file

Taking into account the possible buttons that could be pressed, the UO2 navigation model is represented by the sequence of screens shown in Fig. 5.b (e.g. S0, S11, S12, etc.): the "building plans" button, and the button to move to a next/previous floor plan. Each screen (state) allows the user to touch the building plan in order to move/zoom in on the current floor.

Interface responses are represented by a finite state machine FSM (Fig. 5.c). The pre-process interprets the logs and associates the user activities with both the screen sequence states and the FSM states. The "T0" state corresponds to every possible screen (from S11 to S15). The new states file is easier to read, understand, and trace (see Fig. 5.d). Each line describes the interaction state in terms of three data points: state (screen and touch) and time (seconds and milliseconds). In the example, (1) the S0 state corresponds to the main screen of the FindMyPlace application, (2) S11 is the state reached after pressing the "building plans" button, and the next lines represent (3) a series of six touch activities (Ti). These are: placing the 1st finger on the screen (T0—0→T1), placing the 2nd finger on the screen (T1—261→T2), zooming in on an area of the floor plan (T2—2→T2), removing the 2nd finger from the screen (T2—

262/6→T3), moving the 2nd finger across the screen without effects on the interface (T3—2→T3), and removing the 1st finger (T3—1→T0).

After finishing the evaluations and filtering the log files, we obtained the relevant performance data: (1) efficacy was estimated by considering the percentage of task completion and the number of erroneous user interactions, and (2) efficiency was estimated by considering the performance time for each task and its deviation from the optimal routes according to certain given patterns. Tab. 5 shows the mean of the gathered data. All users completed all tasks, but in some cases users were lost for a moment because they did not know what kinds of interactions could be performed. The time spent to complete the tasks was clearly low. Just one task lasted for over 30 seconds (mean value). The number of interactions performed seems to be high, as it goes from a mean of 25.33 to 49.67, but we think this is justified by the fact that locating a place in building plans when using a small screen requires many movements and zoom interactions. Finally, erroneous interactions were very low; only 6 users made interactions outside of the optimal route. They completed 15 erroneous interactions out of a total of the 1738 interactions performed by all users to complete the whole task set.

Task	Time (ms)	N. interaction	Completion	Erroneous interaction
1	26809.58	33.58	1	0.17
2	27341.25	36.25	1	0.75
3	22611.00	25.33	1	0.25
4	34925.00	49.67	1	0.08

Table 5. Mean values of efficacy and efficiency measures derived from the logs studied

In general, the results of the satisfaction questionnaire were positive. The users considered statements 9 and 10, which concerned error messages, as not applicable, since the interactions of most of them had did not give rise to error messages. On the other hand, 9 statements were ranked between 8 and 9; they included aspects related to how easily the system can be used, the usefulness of the application, comfort and the user's perceived ability to complete the tasks. The two statements regarding the clarity of information and how easy it was to locate information were ranked with a mean of 7. Users ranked three questions with a mean value of between 6 and 7: the clarity of how information was organized on the screens, whether the information provided was valuable for completing the tasks, and whether the user interface was appealing. Comfort when using the interface was ranked at 5.75. Statement 18, "*This system includes all functionalities and capabilities I expect it to have*", was the lowest ranked, although users were told that the evaluation covered just a part of the system. Statement 19, which ranked the user's overall satisfaction with the system, received an average value of 6.67 out of 9.

Users' comments included in the questionnaire were complemented with brief interviews. Some interesting points arose from both techniques. First, in some cases users were not aware of which floor they were on. Although the floor indicator is text displayed on the top of the interface next to the home icon, it was not recognized as part of the interface. Another relevant comment requested colour coding for different kinds of spaces (e.g. offices, classrooms or laboratories); it would have been helpful for identifying the spaces and making the search quicker.

5 Discussion

Applying usability evaluation techniques in the early stages of the development process has shown two main advantages. First, obtaining the initial list of UOs at the project's beginning, together with the continuous evaluation of usability with end users, have promoted development that is tailored to users' expectations. Second, using paper prototyping prior to implementation, when it is used in early stages of the development process and it is focused on checking navigation and layout, have proved to be adequate for validating the Requirement Models. Although FindMyPlace includes interactions such as map zooming or direct manipulation, our results contradict references stating that paper prototyping is not adequate for developing mobile phone applications [Kangas 05].

Thanks to the early evaluation of the SE-HCI model, we were able to detect early on that UO1 required a major iterative redesigning process, and thus we organized the process accordingly. At the same time, we could confirm that performing evaluations with end users at every step of the process, such as [Mayhew 99] suggests, would be outside the scope of agile methodologies, since the time required to define and prepare user tests goes against the agile philosophy. Thus, there must be a balance between agility and evaluation in order to mix both aspects in project development.

Regarding the later steps in the development process, the use of heuristic evaluation provided us with limited information. In the context of InterMod, heuristic evaluation was applied only for developments related to specific UOs. The selected set of heuristics [Bertini 08] was shown to be valuable. However, as they were developed prior to the popularity of multi-tactile devices, they do not deal with the aspects that would have benefitted the evaluation.

Another relevant issue concerning evaluations lies in how to collect data in mobile environments. We have developed a module to analyse the interaction logs resulting from user tests. This module identifies the screen that the user is on at every moment, and extracts information about what activities (moving or zooming) are being performed to explore the building plans. We expect this module be particularly useful for further studies in mobile developments.

Our experience when developing software with InterMod suggests that most evaluations should be carried out by a multidisciplinary team composed of end users and members of the development team. We believe it is desirable to include at least one specialist in usability testing aimed to design and streamline a suitable assessment process, and also to determine the evaluation frequency with end users. Usability evaluations must be performed because these processes ensure better decision-making during development processes, and they avoid wasting time on wrong paths and their subsequent correction. Thus, adequate monitoring of the project without causing end user tedium is a challenge.

By considering the assessment needs with end users it is possible to plan the parallel creation of models by different UOs in the same iteration. Thus, usability evaluation sessions with end users could be grouped without delaying the project's development. As a direct consequence, the time users engage in evaluation would be reduced and the evaluation criteria could be unified. For instance, the end user evaluations of the M-1 models were performed jointly in iteration2. Thanks to the Internet and new technologies, certain types of user evaluations, such as completing questionnaires, may not require direct user observation. Nevertheless, evaluations with the prototypes should be carried out in their presence in order to gather opinions, ideas and impressions. The most frequent criticism of agile methodologies is their lack of formality, as obtaining a product seems to have priority over the time invested in its design. However, the SE-HCI model proposed by InterMod includes the formalization of the gathered requirements and also promotes early evaluation.

InterMod also provides solutions for adequately integrating usability in an agile development methodology. The traditional agile methodologies that do not consider usability aspects have problems regarding when and how to integrate them in the development process. These problems arise from the fact that agile models boost short iterations and minimal up-front design. It must be taken into account that, in this context, changes in the development process can happen at any stage [Beck 01]. Continuous changes to the user interface (UI) derived from a fast iterative design can cause conflicts with user expectations; they can lead to inconsistencies and dissatisfaction [Constantine, Lockwood 02a]. InterMod's integration proposal shares some of the solutions to such integration problems that have been presented in the literature and its agile process facilitates adequate symbiosis between agility and usability assessment.

6 Conclusions

This paper has presented a proposal to integrate usability engineering in the agile InterMod methodology, and has shown its advantages, benefits and verifiable improvements. The main contributions consist of explicitly describing when, which and how to integrate usability techniques; and they have been progressively presented through the FindMyPlace guiding example. On the one hand, this approach promoted development tailored to users' expectations as a consequence of applying usability evaluation techniques in early stages of the development process. On the other hand, the SE-HCI model proposed by InterMod includes the formalization of the gathered requirements, which provides a level of formality in the agile development and also promotes early evaluation. The performed usability evaluations ensure better decision-making during the development processes and saves time and effort by avoiding wrong paths and their subsequent correction.

InterMod is an agile methodology that aims to help with the accurate development of high quality interactive software. Its main characteristic is to develop interactive software on the basis of models generated and evaluated during the project. The solution provided by InterMod organizes the development process on the basis of UOs, which are logical and consistent units. Each UO considers evaluation processes advocated by HCI, which means that usability design and evaluation are performed throughout the entire development process. The different activities of different UOs are developed in parallel. Thus, the UI usability design and evaluation are not performed up-front for the whole project, as they are framed by UOs. In this context, new integration activities are necessary to evaluate the UI, as it can be enhanced while the development process advances. For the cost of the evaluations to be affordable in an agile development context, the management of UO activities may mean grouping such evaluations in specific iterations in order to achieve a greater degree of efficacy when performing them. The presented work shows that performing such grouping is possible and beneficial for building an agile mobile application.

In terms of mobile development, applying usability evaluation has provided clear advantages from the earliest stages. First of all, paper prototypes allowed us to determine usability problems in one of the UOs. In addition, it has allowed us to focus on developing the remaining UOs while solving the detected problems. Secondly, heuristic evaluations have proven to be adequate for evaluating early mobile prototypes, as they provide a fast and cheap way to determine usability errors. Thirdly, the module developed to analyse user interaction logs has been valuable in checking their efficacy and efficiency. Furthermore, the success of this mobile log analysis module will promote its use in other mobile developments to perform the same function.

Acknowledgments

This work has been supported by the TIN2009-14380 and GV IT722-13 research projects. The authors would like to thank Sergio Jiménez for his helpful suggestions.

References

[Ambler 13] Ambler, S. “Agile Modeling (AM): Effective Practices for Modeling and Documentation”. Retrieved on May 2013 from: http://www.agilemodeling.com/

[Beck 01] Beck, K. 2001. “Manifesto for Agile Software Development”; Agile Alliance. Retrieved on May 2013 from: http://agilemanifesto.org/

[Bertini 08] E. Bertini, T. Catarci, A. Dix, S. Gabrielli S. Kimani, G. Santucci. “Appropriating Heuristic Evaluation for Mobile Computing”; International Journal of Mobile Human Computer Interaction, 1(1), 20-41

[Blomkvist 2005] Blomkvist, S.. “Towards a model for bridging agile development and user-centered design”; In Seffah, A., Gulliksen, J., Desmarais, M. (eds) *Human-Centered Software Engineering- Integrating usability in the Software Development Lifecycle*. Springer, The Netherlands. 2005

[Constantine, Lockwood 02a] Constantine L., Lockwood L. 2002. “Usage-centered engineering for web applications”; IEEE Soft. 19(2):42-50.

[Constantine 02b] Constantine, L. “Beyond User-Centered Design and User Experience: Designing for User Performance”, Cutter IT Journal, vol. 17, no. 2, feb. 2002.

[Dix 03] Dix A., Finlay J. E., Abowd G. D., Beale R. 2003 “Human-Computer Interaction” *3rd ed*. Prentice Hall.

[Dumas, Redish 93] Dumas J.S. and Redish J.C. 1993. “A Practical Guide to Usability Testing”, Norwood.

[Federoff 08] M. Federoff, C. Villamor, L. Miller, J. Patton, A. Rosenstein, K. Baxter, y K. Kelkar, 2008 “Extreme usability: adapting research approaches for agile development”, in CHI ’08 Extended Abstracts on Human Factors in Computing Systems, New York, NY, USA, , pp. 2269–2272.

[Granollers 05] Granollers, T., Lorés, J., Cañas, J.J. 2005. "Design of user centred interactive systems. UOC Editorial (In Spanish)

[Hellman 10] Hellman, T.D., Hosseini-Khayat, A., Maurer,F., "Agile Interaction Design and Test-Driven Development of User Interfaces – A Literature Review", in Agile Software Development, T. Dingsøyr, T. Dybå, y N. B. Moe, Eds. Springer Berlin Heidelberg, 2010, pp. 185–201.

[Hussain 09] Hussain, Z., Wolfgang, S., Holzinger, "A. Current State of Agile User-Centered Design: A Survey", in Proc. of the 5th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society on HCI and Usability for e-Inclusion pp.416 - 427 Springer-Verlag Berlin, Heidelberg, 2009.

[ISO/IEC 9126-1 01] International standard, "Software engineering—Product quality—Part 1: Quality model*"*.

[Kaikkonen 05] Kaikkonen, A., Kallio, T., Kekäläinen, A., Kankainen, A. and Cankar, M. 2005. "Usability testing of mobile applications: A comparison between laboratory and field testing"; Journal of Usability Studies, 1(1), 4-16.

[Kangas 05] Kangas E. Kinnunen T. "Applying user-centered design to mobile application development" ACM 48(7): 55-59, 2005

[Kane 03] Kane, D., "Finding a Place for discount usability engineering in agile development: Throwing down the gauntlet";in proceedings of Agile Development Conference (ADC'03), pp. 40-46. IEEE Press, 2003

[Larman 03] Larman, C. 2003. "Agile and Iterative Development: A Manager's Guide" Addison-Wesley Professional

[Lee 06] Lee, J.C. 2006. "Embracing agile development of usable software systems"; In CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06). ACM, New York, NY, USA, 1767-1770

[Lee 09] Lee, J.C.; Scott McCrickard, D.; Stevens, K.T.; "Examining the Foundations of Agile Usability with eXtreme Scenario-Based Design" AGILE '09, pp.3-10, 24-28 Aug. 2009

[Lewis 82] Lewis, C. H. 1982. "Using the "Thinking Aloud" Method In Cognitive Interface Design" (TR). IBM. RC-9265

[Lewis 01] Lewis, J. R. 2001. "Psychometric Evaluation of the CSUQ Using Data from Five Years of Usability Studie*"*;. IBM. TR-29.3418

[Limbourg 05] Limbourg Q., Vanderdonckt V., Michotte B., Bouillon L. 2005. "USIXML: A Language Supporting Multi-path Development of User Interfaces" LNCS , 3425, 200-220.

[Losada 09] Losada, B., Urretavizcaya, M., Fernández- Castro, I. 2009. "Requirements analysis as a guide for the process of organising and developing an interactive application*"*. Int. Ass. Development of the Information Society, pp. 412-416.

[Losada 12a] Losada B, Urretavizcaya M., Fernández-Castro I., 2012. "A guide to agile development of interactive software with a "User Objectives"-driven methodology", Science of Computer Programming, DOI 10.1016/j.scico.2012.07.022. In press.

[Losada 12b] Losada B, Urretavizcaya M., Fernández-Castro I., 2012. "Combining InterMod agile methodology with usability engineering in a mobile application development"t, *Interacción'12. DOI: 10.1145/2379636.2379674*

[Löwgren, Stolterman 07] Löwgren J., Stolterman J., 2007 "Thoughtful Interaction Design: A Design Perspective on Information Technology". The MIT Press.

[Markopoulus 00] Markopoulus P., Marijnissen P., 2000 "UML as a representation for Interaction Design". OZCHI, 240-249

[Mayhew 99] Mayhew, D.J. 1999. “The Usability Engineering Lifecycle: A Practitioner’s Handbook for User Interface Design”. Academic Press

[Memmel 07] Memmel T., Gundelsweiler,F, Reiterer,H. 2007 “Agile human-centered software engineering” in Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it - Volume 1, Swinton, UK, UK, pp. 167–175.

[Nielsen 93] Nielsen, J. 1993. “Usability Engineering”, Academic Press.

[Nielsen, Mack 94] Nielsen, J. and Mack, R. 1994. “Usability Inspection Methods”; Wiley&Sons.

[Nielsen, Molich 90] Nielsen, J., Molich, R. 1990. Heuristic evaluation of user interfaces, Proc. ACM CHI'90 Conf. pp. 249-256

[Nielsen 11] Nielsen, J. “Usability of Mobile Websites & Applications: 237 Design Guidelines for Improving the User Experience of Mobile Sites and Apps”. Nielsen Norman Group. 2nd edition. Available online at: http://www.nngroup.com/reports/mobile/

[Nunes 00] Nunes, N., Cunha, J.F. 2000 “Towards a UML Profile for Interactive Systems Development: The Wisdom Approac”. In Andy Evans, ed. Proceedings of the International Conference on the Unified Modeling Language, UML’00, pp 101-116.

[OMG 03] Object Management Group (OMG). “Model Driven architecture. Technical report”, 2003. Retrieved on May 2013 from: http://www.omg.org/mda.

[Obendorf 08] Obendorf H., Fink, H. “Scenario-based usability engineering techniques in agile development processes”; In CHI '08 Extended Abstracts on Human Factors in Computing Systems (CHI EA '08), ACM Press, 2008, pp. 2159-2166

[Paternò 99] Paternò, F. “Model-Based Design and Evaluation of Interactive Applications”; Springer-Verlag London. 1999

[Pinheiro 02] Pinheiro da Silva, P. “Object Modelling of interactive systems: The UMLi Approach”. Ph. Thesis 2002.

[Ratcliffe 11] Ratcliffe, L., McNeill, M. “Agile Experience Design: A Digital Designer's Guide to Agile, Lean, and Continuous (Voices That Matter)”; New Riders. 2011

[Robles 10] Robles, E., Panach,J., Grigera, J., Rossi,G., Pastor, O. “Incorporating usability requirements in a test/model-driven web engineering approach”; J. Web Eng. 9, 2 (June 2010), 132-156.

[Singh 08] Singh, M. “U-SCRUM: An Agile Methodology for Promoting Usability”; Agile, 2008. AGILE '08. Conference , vol., no., pp.555-560, 4-8 Aug. 2008 doi: 10.1109/Agile.2008.33

[Stone 05] Stone, D.C., Jarrett, M., Woodroffe, S., Minocha. “User Interface Design and Evaluation”; Morgan Kaufmann 2005

[Sutcliffe 05] A. G. Sutcliffe, “Convergence or Competition between Software Engineering and Human Computer Interaction”; in Human-Centered Software Engineering, vol. 8, A. Seffah, J. Gulliksen, y M. C. Desmarais, Eds. Springer-Verlag, pp. 71–84., 2005

[Sy 07] Sy D. 2007. “Adapting Usability Investigations for Agile User-Centered Design”; J. Usability Studies, 2(3): 112-132.

[Wharton 94] Wharton C., Rieman J., Lewis C. “The cognitive walkthrough method: a practitioner's guide. In Usability Inspection Methods” J. Nielsen & R. Mack. pp. 105-140 John Wiley & Sons, 1994.