

# Framework for Integrating Usability Practices into the Software Process

Xavier Ferre, Natalia Juristo, and Ana M. Moreno

Universidad Politecnica de Madrid  
Campus de Montegancedo  
28660 - Boadilla del Monte (Madrid), Spain  
{xavier, natalia, ammoreno}@fi.upm.es  
<http://www.ls.fi.upm.es/udis/>

**Abstract.** Software development organizations wanting to introduce usability practices into their defined software process have to undertake laborious efforts for that purpose, since, for the time being, there exists a lack of reference model or framework which indicates where and how in the software process usability needs to be considered. They also have to overcome the important differences between HCI (Human-Computer Interaction) and SE (Software Engineering) in terminology and approach to process definition. We offer developers who have the objective of integrating usability practices into their software process, a framework that characterizes 35 selected HCI techniques in relation to six relevant criteria from a SE viewpoint, and organizes them according to the kind of activities in the development process where they may be applied, and to the best moment of application in an iterative life cycle. The only requirement for the existing software process is to be based on an iterative approach.

## 1 Introduction

Usability has been present in software quality attribute decompositions since the late 70s [1], but it has been recently that it has begun to appear as a highly relevant attribute for customer-perceived software quality. For managers, for example, usability is a major decision factor, particularly for selecting a product [2]. Emerging fields like web development have contributed to this increased impact of usability into software development. According to Donahue, having a competitive edge in usability is crucial for e-commerce sites [3]. But not only web development is concerned by usability, it is relevant for any interactive software system with human users.

The HCI field has been pursuing the development of usable software products for a long time. HCI offers a large number of techniques which may be applied for the aim of producing usable software. They are widely applied in

development projects where usability is the main (or only) quality attribute taken care of. But the application of HCI techniques is decoupled from the overall software development process.

This state of things is changing. Recently, software developers are becoming aware of the importance of usability issues [4]. A clear sign of the importance that usability issues and their integration with the rest of development activities is gaining, lies in the newly created Usability Process in the first amendment to the ISO/IEC Standard 12207 for Software Life Cycle Processes [5]. The first activity in the Usability Process deals with specifying how user-centered activities fit into the whole system lifecycle process, and to select usability methods and techniques. In this way, usability practices integration into software the process is no more an area of interest for some specialists, but an endeavor that matters for the SE software process area.

In despite of that, software development organizations wanting to properly manage the usability level of their software products, still face a real challenge when dealing with the integration of HCI techniques into their defined software development process [6]. Current SE practice does not correctly address usability issues throughout development. Developers sometimes wrongly consider usability as being just related to the design of the visual elements of the GUI, and therefore it is dealt with late in the development. On the contrary, a proper user-centered development (the HCI approach to software construction) deals with usability issues all over development. In particular, observing users and checking that their needs and likings are considered early in the process is crucial for the production of usable software. On top of that, usability integration into the process is hindered by the differences in terminology and development philosophies between HCI and SE [2] [7]. HCI literature does not offer to organizations that follow a strong SE approach a ready-to-use manual for HCI technique application in a SE development process. Some HCI techniques are not allocated to particular activities in the process, and the activities terminology varies between HCI authors [8]. As a result, the effort of interpreting HCI literature from a SE perspective presents an added difficulty to the task of usability practices integration.

Our approach is to offer software developers a selection of HCI techniques which are more appropriate to be incorporated into a defined software process. The HCI techniques are integrated into a framework organized according to the kind of SE activities and to the moment in an iterative development where their application yields a higher usability improvement. The characterization of HCI techniques is based on a SE perspective, in order to use a terminology which is familiar to developers. The framework aims to be a flexible enough tool so that a particular process model is not required for its application. The only requirement for the existing development process is to be based on iterative refinement, since it is necessary characteristic of any user-centered development effort. Our general approach to HCI integration into SE development processes is discussed in [9], along with a review of other existing proposals to the integration issue.

The framework is aimed to software development organizations with a strong SE background, which have identified the importance of usability and want to enhance their software process with HCI techniques and activities. The selection of HCI techniques for inclusion in the framework responds to these organizations needs. For example, it favors techniques which require less training for a software engineer, and techniques that yield a higher improvement in the usability of the product compared to the application effort.

The research presented in this work has been carried out as part of the project STATUS, financed by the European Commission (IST - 2001 - 32298). Two software development companies were partners in the project consortium, and they participated in the definition and refinement of the framework presented here. They provided us with input about real industrial needs regarding usability practices integration into the software process.

For the definition of the framework it has been necessary to perform a study on HCI activities and techniques in the first place, and to carry out a selection of techniques for the framework. This study is summarized in section 2. Once selected, techniques have been assigned to SE activities in section 3, so that developers know which kind of existing activities some techniques correspond to, and which additional activities need to be added to the process for properly handling usability. We have also performed a study on the conditions to be met by a software process to be considered for usability practices integration. Additionally, as there are some indications in the HCI literature about the best moment of application for a certain number of techniques, as detailed in section 4, we have considered as well the moment of application of the selected HCI techniques in an iterative development. The results are packaged for developers in a framework that offers three views, and section 5 details the usage of the framework. Section 6 describes the conclusions reached. Finally, the framework in its view by development activities is included as an appendix.

## **2 Selection of HCI Techniques**

Whereas one of the goals of SE has been to formalize the process to make software development systematic and disciplined, HCI does not take a similar formalized view of the development. HCI concerns include, on top of software systems, their interaction with the user and with the social environment where system-user interaction takes place. The complexity of these issues has led HCI to focus on certain development issues, and to consider some issues with less detail than SE. That is the case with process issues. Each author in the HCI field has a particular view of the development in terms of what activities should be undertaken to produce a usable software product, and techniques for this purpose are not always described associated to a particular activity. Therefore, it has been necessary for our purpose of usability practices integration into the software

process, not only to survey HCI literature for proposed activities and techniques, but also an effort to clarify some activities with different names, different activities with similar names and an assignment of techniques to activities. In this way, we have been able to get well-established practices redefined into concepts closer to SE that allow relating them with the software process. The survey on HCI activities and the classification of the identified 95 HCI techniques into such activities is detailed in [7].

The number of techniques identified in HCI literature (95) is too large for a developer to consider when aiming usability practices integration into the development process, so we have performed a selection of the techniques that, for one reason or another, seem to be more appropriate for the integration purpose. In order to evaluate the appropriateness, we have characterized the HCI techniques according to the following criteria:

- **User Participation:** One of the basic points of a user-centered approach is the active involvement of the future system users. Some usability techniques are specifically designed to encourage this involvement. Since the goal of this work is to facilitate the use of usability practices during development, and user involvement is one of the main and critic of these practices, techniques that specially support user involvement in development will be preferred for integration into the software process.
- **Training Needs:** This criterion refers to how much training average software engineers would need to be able to apply the technique with any chance of success. Techniques which require less training will be less costly to adopt for organizations with little previous HCI experience.
- **General Applicability:** This criterion reflects the scope of the technique, that is, how applicable it is to a wide range of software development projects. Some HCI techniques, for example, are aimed to studying expert user performance in optimal conditions of usage, which only a fraction of software systems may consider the most relevant usability attribute (for instance, a pilot cockpit system). Since we are aiming to cover a big variety of development projects, we will select first wide-ranging techniques.
- **Proximity to SE:** This criterion reflects whether the principles on which the technique is based match the principles and approaches usually present in SE. Techniques with a high value in this criterion will be less difficult to introduce in an organization with a strong SE background.
- **Usability Improvement/Effort Ratio:** This criterion refers to how much the use of the technique can improve the usability of the final product compared to the effort involved in its application. As resources are usually scarce in many software development projects, we believe that cost/benefit information can be very useful for technique selection.
- **Representativeness:** This criterion reflects how commonly the technique is applied in the field of HCI. As an indicator of this criterion, we are going to

use the number of consulted authors who recommend the technique. We will favor in technique selection the ones that reach a higher consensus.

The values for these six criteria have been assigned to each HCI technique based on a SE perspective, aiming to ease the integration by reducing the number of techniques for consideration. The complete characterization is detailed in [10].

As the goal is to select techniques, we have summarized the combined value of each criterion in an additional summary criterion called **Total Rating**. This criterion can adopt the following values:

- **Very useful:** Techniques that are especially useful for our purpose because their usability improvement/effort ratio is high, they do not require a lot of training, they are within the medium to high applicability range, and they are either very close to SE or are techniques that are commonly applied in the HCI field.
- **Useful:** Whereas the techniques that meet the above conditions are the best for the aim we pursue, other techniques may also be useful because they either have a high usability improvement/effort ratio (which is considered useful just by itself), or it is medium but they have besides either a medium to high applicability or they encourage user participation.
- **Not very useful:** Techniques that do not meet the requirements for belonging to either of the above categories have been labeled as not very useful in the total rating. Although they may be useful for some development projects, we do not think that this is enough to merit the effort of integrating them into the software process. Notice that the goal of this work is not the inclusion of the whole HCI field into the development, so not all the existent techniques should be selected.

The proposed framework is not designed as a solution that has to be adopted as a whole, because it will include a range of techniques that could be useful in some cases. Additionally, one goal we aim to is to keep the complexity of the framework within reason, in order to enhance comprehensibility and applicability from the viewpoint of SE. Accordingly, we have decided to select both techniques whose total rating is “very useful” and techniques for which the total rating is “useful”, as the resulting set of techniques (35) achieves the above goals. The characterization of the selected techniques is detailed in the framework view presented in the appendix.

### 3 Mapping of HCI Techniques to Software Process Activities

The main purpose of this work, apart from offering a set of selected HCI techniques, is to characterize them according to software process terminology from the SE field, so they can be naturally integrated into the software process. With this aim, we have first studied the correspondence between HCI activities and SE activities, to identify which kind of activities in the software process will

include HCI techniques, and then we have mapped the individual selected techniques to such kind of activities in the software process.

### **3.1 Software Process Activities Affected by Usability**

When approaching HCI activities we identify that there are two types of activities with regard to SE: The ones that share objectives and, therefore, should be integrated, and the ones that deal with issues that are not commonly dealt with in SE processes. When mapping HCI activities to their respective SE ones, IEEE Standard 1074-1997 [11] activity group names have been used for describing high-level process activities grouping. Nevertheless, this work aims to be useful for any organization with a defined software process, regardless of the particular approach taken for developing software life cycle processes. Therefore, we have not adhered to a particular software process standard, because our recommendations regarding usability practices integration into the software process do not require the organization compliance with any standard. For this reason, we have used generally accepted SE terminology for the classification of software process activities, instead of using activity names from a specific software process standard. The mapping between HCI and SE activities is represented in Figure 1.

The greatest degree of similarity in objectives lies in requirements activities, since they overlap with the concerns in HCI about specifying the context of use, analyzing users and their tasks, specifying usability requirements, developing a product concept, producing low-fidelity prototypes, and validating them with representative users. As a result, existing requirements-related activities of the software process may be enhanced by the introduction of HCI techniques and approach.

Other HCI activities, not related to SE requirements activities, do not have a direct coincidence of objectives with existing SE activities, so they have to be added to the existing development process for it to produce usable software. These new kinds of activities are Interaction Design and Usability Evaluation.

Interaction Design is concerned with the definition of the interaction environments and their behavior, including the coordination of the user-system interaction. It also includes the design of the visual elements making up the GUI (Graphical User Interface) and their behavior, when the user interface is graphical. Interaction Design gathers a kind of activities with a certain degree of independence from activities in the Design activity group, but it is related to the sort of tasks carried out in such activities. Therefore, we have considered that Interaction Design belongs to the Design activity group, but with its own place inside it.

Usability Evaluation includes all the activities that aim to evaluate the usability of the product being developed. Consequently, they can fit in the Evaluation activity group. Nevertheless, they are quite independent from other

activities in the group because usability practices for evaluation differ to a great extent from SE evaluation practices. Therefore, we have gathered all kinds of usability evaluation practices in Usability Evaluation, and placed this kind of activities inside the Evaluation activity group. The typical HCI decomposition of Usability Evaluation considers three types of activities: Expert Evaluation, Usability Testing, and Follow-Up Studies of Installed Systems.

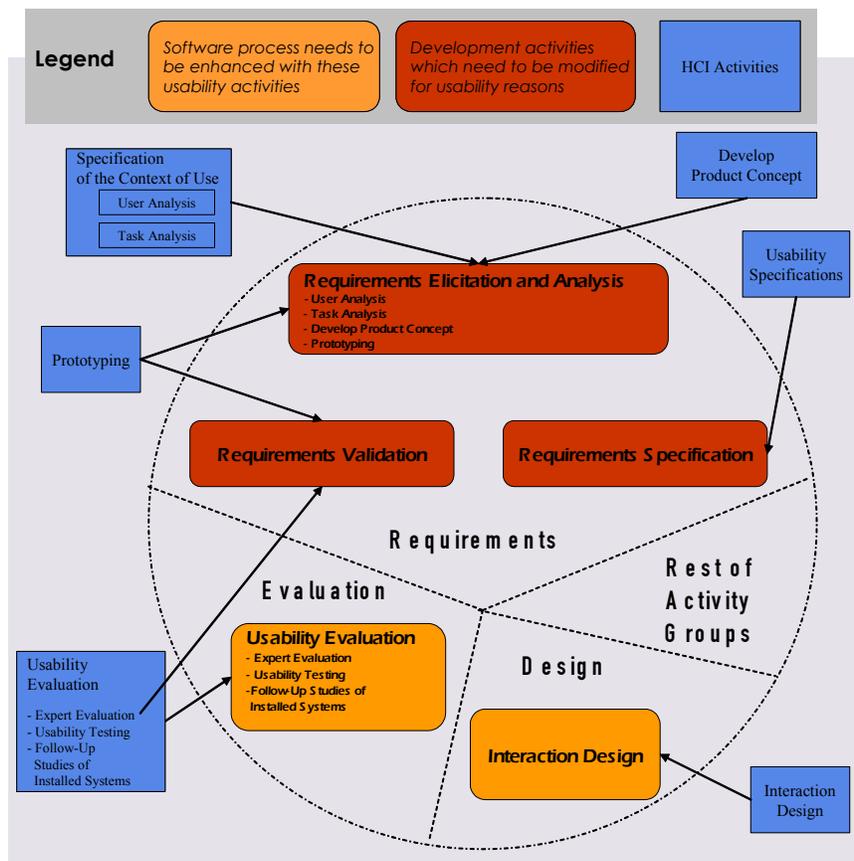


Fig. 1. Mapping of HCI Activities to SE Activities in the Software Process

### 3.2 Assignment of HCI Techniques to Activities

The previous mapping between HCI and SE activities represents the main basis for assignment of HCI techniques to kinds of activities into the software process. But, due to the above mentioned lack of formality in process issues in the HCI

field, not every HCI technique is clearly assigned to an HCI activity, and terminology varies between authors, so the mapping between HCI and SE activities is not enough for the assignment of HCI techniques to activities. Therefore, in addition to the mapping between activities, we have considered the particularities of each technique, as described in the HCI literature, for its assignment to a specific kind of activity. Between all the activities in a software process, we have just considered the ones affected by usability, as seen in the previous section.

For existing activities in development that need to be modified with the inclusion of HCI techniques, we have kept SE terminology. Nevertheless, for Requirements Elicitation and Analysis activities we have identified 15 HCI techniques that can be applied when performing such activities. The large number of HCI selected techniques assigned to Requirements Elicitation and Analysis activities is due to the great importance attached in the HCI field to observing the users, analyzing their characteristics and tasks, and developing the right product concept. Since 15 is quite a high number of techniques to present in a single set, we have decided to structure them according to the HCI activity into which they are applied in HCI methods (for techniques which are actually assigned to an activity). In this way, we aim to improve communication between usability specialists and developers in the critical activities of Requirements Elicitation and Analysis. They are the kind of activities which require a greater degree of integration of current SE practices with HCI ones, and, therefore, they present a greater need of communication inside joint SE-HCI teams.

The final results of the framework detailed in the appendix show the resulting organization of techniques according to the kind of development activities where they apply. A more thorough discussion of the reasons for assigning each technique to a particular kind of activity, along with a more detailed description of the mapping between HCI and SE activities, can be found in [10].

## **4 Embedding HCI Techniques in an Iterative Life Cycle**

We have studied the HCI literature to identify the characteristics that software development should have for it to be considered user-centered. [12], [13], [14], [15], and [16] agree on considering iterative development as a must for a user-centered development approach. The complexity of the human side in human-computer interaction makes it almost impossible to create a correct design at the first go. Cognitive, sociological, educational, physical and emotional issues may play an important role in any user-system interaction, and an iterative approach is the most sensible way to deal with these issues.

The other three characteristics that are mentioned by several sources are: active user involvement; a proper understanding of user and task requirements; and multidisciplinary knowledge. The three conditions can be met by introducing

HCI practices into the software process. They may contribute with knowledge from other disciplines, they may also offer participatory ways for software developers to integrate users into the design process, and they may enhance requirements activities with specific usability aspects as well. On the contrary, the first condition (that is, to be based on an iterative approach), is an intrinsic characteristic of the process, and it will be the only requirement for a defined software process to be a candidate for usability practices integration through the proposed framework.

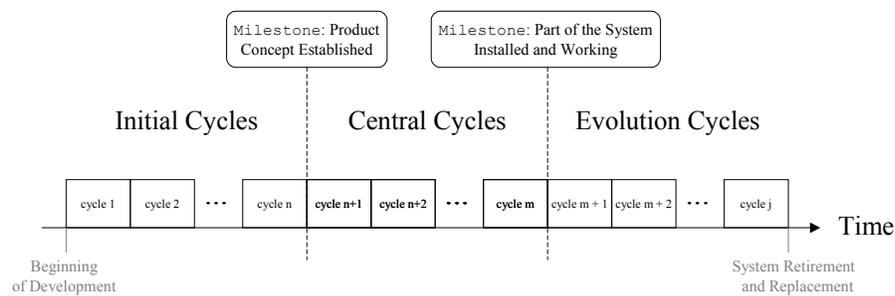
There are indications in the HCI literature on when in development time each technique yields the most useful results for improving the usability of the final product. In particular, HCI literature favors an early application of certain HCI techniques, while others yield useful when a working system is installed in the user organization.

An iterative life cycle is made up of cycles, but not all the cycles are the same. Each cycle implies working in different kinds of activities, and the relative effort and emphasis change over time [17]. During the first development cycles, a greater effort is dedicated to activities related to problem delimitation and to establishing the solution outline, while, as development advances, design and implementation activities gain more weight. Therefore, different stages can be defined in an iterative life cycle.

On top of the assignment of HCI techniques to activities, we consider necessary to offer developers a guide about the more appropriate moments in development time for the application of each HCI technique. We aim to transmit this kind of information present in HCI literature to developers. Since each organization has its own terminology for phases and milestones, we have established a generic representation of moments in iterative development time, which condenses the most important stages for HCI technique characterization. We have considered the following three stages: Initial Cycles, Central Cycles and Evolution Cycles. Development time is roughly divided into these three stages by the following two milestones:

- **Product Concept Established:** In any iterative process, the early cycles are given over to the points identified as being the riskiest, and one of the biggest risks as far as usability is concerned is not having in mind the right product concept, the one that fits the user experience and expectations. Some HCI techniques focus on the production and/or evaluation of early tentative designs and, therefore, they are especially well suited for application in the Initial Cycles, which account for the cycles before the product concept has been established. Central Cycles begin once this milestone has been reached.
- **Some Part of the System Installed and Working:** Some usability evaluation techniques cannot be applied unless a version of the software system under development has been installed at the end user's workplace. Evolution cycles begin once this milestone has been reached.

Figure 2 shows this division of development time in stages. Developers need to match this generic representation to the terminology used in their organization, in order to interpret our proposed framework. We have graded every selected HCI technique according to their suitability for each stage described in the consulted literature, with one of the following values: **Especially well-matched** (meaning that the technique will be of utmost usefulness when applied in the development stage in question); **Neutral** (the technique is applicable in this stage, although this stage does not stand out as being any better matched than other development times); and **Not usual** (it is not the more appropriate stage for this technique application and there are other more appropriate techniques).



**Fig. 2.** Division of Iterative Cycles into Stages

The framework view detailed in the appendix includes the characterization of selected HCI techniques according to moments in an iterative life cycle. A detailed discussion of the reasons for grading each HCI technique with respect to their suitability to each development stage can be found in [10].

## 5 Integrating Usability Practices into the Software Process: A Framework

We have packed into a framework the characterization of HCI techniques detailed in previous sections. The framework allows developers to take decisions about the specific HCI techniques to include into their software process in order to improve the usability of the developed software product. It condenses all the information developers need to apply the solution we are proposing.

In order to ease the usage of the framework by its potential users, we offer developers three views:

- **View by HCI Techniques:** This view is organized by the name of the selected HCI techniques. It yields useful when developers already know about a particular HCI technique they have heard about, or whose usability benefits they have been convinced of. Accessing the framework through

this view offers the possibility of knowing the characterization of a particular HCI technique.

- **View by Development Activities:** When developers are trying to find appropriate HCI techniques for activities in their process they access the framework through this view. In order to use the framework with this purpose, developers need to translate the generic activities in the framework to the specific activity names considered in their organization. Then, between the set of techniques suggested for each kind of activity, they may select those ones that better fit their particular objectives.
- **View by Moments of Application:** An organization of the selected HCI techniques according to stages in the development life cycle is offered to developers through this framework view. For example, an organization may have identified that they need to pay more attention to usability in the earlier stages of development, and wants to know which HCI techniques are more suitable for application at this stage. In order to use the framework through this view, developers need to fit the generic representation of stages in an iterative cycle used in the framework to the phases or stages considered in their organization.

The three views offered in the framework are not exclusive. They may be used independently, but the expected usage of the framework is by means of an information search that combines several views. For the task of integrating usability practices into the software process, developers may switch between views for reaching the kind of information they are dealing with at each moment. For example, when dealing with the introduction of HCI practices into a particular activity, the developer will make use of the view by activities, by matching his or her organization activity name with the corresponding kind of activity in the framework. Between the techniques proposed for such kind of activity, the developer will choose one or more techniques, considering both the criteria characterization and the information about the better moment of application of each technique. Then, the developer may change to the view by moments of application, to know about the techniques which are bound to be applied at the same moment in development time as the techniques selected. Finally, the developer may switch to the view by HCI techniques to look up a technique which he has already placed in the process, because he or she wants to remember the training needs of the technique, in order to know if some members of the development team will need to attend a lengthy usability training course.

We consider that these different views of the framework offer a flexible tool for easing the usability integration endeavor. In order to provide some more guidance to the user of the framework, techniques labeled as "Very useful" are highlighted in every view by presenting them as the first ones in each category. These techniques are suggested to the user as the best generic choice. Nevertheless, the developer may take into account the specific constraints of the organization or the project at hand, and, considering the characterization criteria for the techniques, he or she may choose a set of techniques which fit better his

or her needs. The framework is not conceived as a tool for automatic generation of a custom process add-in, but as a tool for supporting the developer in the difficult task of integrating HCI practices into the software process. For this purpose, it condenses some knowledge about the HCI field in a way understandable by developers with a SE background.

Knowing where to plug HCI techniques and activities into the existing process does not automatically make software engineers capable of integrating HCI practices into their day-to-day work. We offer in the framework a basic reference for each HCI technique where developers may turn to if they need more detailed information on the application of the technique, but the application of the framework needs to be supplemented by appropriate usability training for developers. A basic catalogue of HCI techniques, with their main references and a training plan, can be found in [18].

The Appendix details the framework in its view by development activities.

## **6 Conclusions and Future Directions**

Usability practices integration into the software process is a difficult issue due to the existing breach between SE and HCI. Both fields have a different approach to development, different terminology, and different view of process issues. HCI literature is not very helpful for the integration aim, since the information about techniques is not presented in terms a software engineer can easily understand.

We have presented a framework for usability practices integration into the software process that defines which kind of development activities are affected by usability, and characterizes HCI techniques according to such activities and according to the moments in an iterative cycle where their application yields more useful. Additionally, it characterizes each HCI technique according to six criteria which may be of interest to developers in the usability integration endeavor.

The framework has been applied by STATUS partners after a 24 hours (one week) usability course, with good results: Integration of usability has improved requirements understanding, and the usability level of software products developed has increased substantially (including user satisfaction). As a global valuation, developers consider that the effort employed in the application of HCI techniques pays off, considering the results obtained. They were previously unaware of the possible relationship of usability with their development practices, and they do know now where to look for HCI techniques to apply. They are also more aware of the criticality of usability in software development.

We have released a prototype of a web-based tool for easing the application of the framework [19]. With the release of the framework in the web, we expect to refine it with the feedback received from developers and usability specialists who may use it for their particular needs.

## References

1. McCall, J.A., Richards, P.K., and Walters, G.F.: Factors in Software Quality. vol. 1, 2, and 3, AD/A-049-014/015/055, National Tech. Information Service (1977)
2. Seffah, A., Metzker, E.: The Obstacles and Myths of Usability and Software Engineering. *Communications of the ACM*, Vol. 47, no. 12 (December 2004) 71-76
3. Donahue, G.M.: Usability and the Bottom Line. *IEEE Software*, Vol. 18, no. 1 (January/February 2001) 31-37
4. Juristo, N., Windl, H., Constantine, L.: Introducing Usability. Guest Editor's Introduction. *IEEE Software Special Issue on Usability Engineering*, Vol. 18, no. 1 (January/February 2001) 20-21
5. ISO/IEC. International Standard: Information Technology. Software Life Cycle Processes. Amendment 1. ISO/IEC Standard 12207:1995/Amd.1:2002. ISO, Geneva, Switzerland (2002)
6. Ferre, X., Juristo, N., Windl, H., Constantine, L.: Usability Basics for Software Developers. *IEEE Software*, Vol. 18, no. 1 (January/February 2001) 20-29
7. Ferre, X., Juristo, N., Moreno, A.M.: Improving Software Engineering Practice with HCI Aspects. *Software Engineering Research and Applications - Lecture Notes in Computer Science*. Vol 3026, Springer (2004) 349-363
8. Ferre X., Juristo, N., Moreno. A.M.: Deliverable D.5.1. Selection of the Software Process and the Usability Techniques for Consideration. STATUS Project (2002) <http://www.ls.fi.upm.es/status/results/deliverables.html>
9. Ferre, X., Juristo, N., Moreno, A.M.: Integration of HCI Practices into Software Engineering Development Processes: Pending Issues. In: Ghaoui, C. (ed.): *Encyclopedia of Human Computer Interaction*. Idea Group Reference (2005)
10. Ferre, X., Juristo, N., Moreno, A.M.: Deliverable D.6.6 Final Results on the Integrated Software Process. STATUS Project (2004) <http://www.ls.fi.upm.es/status/results/deliverables.html>
11. IEEE: IEEE Std 1074-1997. IEEE Standard 1074 for Developing Software Life Cycle Processes. IEEE, New York (NY), USA (1998)
12. ISO: International Standard: Human-Centered Design Processes for Interactive Systems, ISO Standard 13407: 1999. ISO, Geneva, Switzerland (1999)
13. Constantine, L. L., and Lockwood, L. A. D.: *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*. Addison-Wesley, New York (NY), USA (1999)
14. Shneiderman. B.: *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, Reading (MA), USA (1998)
15. Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey. T.: *Human-Computer Interaction*. Addison Wesley, Harlow, England (1994)
16. Hix, D., and Hartson, H.R.: *Developing User Interfaces: Ensuring Usability Through Product and Process*. John Wiley and Sons, New York (NY), USA (1993)
17. Larman, C.: *Agile and Iterative Development. A Manager's Guide*. Addison-Wesley, Boston (MA), USA (2004)
18. Ferre, X., Juristo, N. and Moreno A.M.: Deliverable D.5.2. Specification of the Software Process with Integrated Usability Techniques. STATUS Project (2002) <http://www.ls.fi.upm.es/status/results/deliverables.html>
19. Ferre, X. *Web-Based Framework for Integrating Usability Practices into the Software Process* (2005) <http://www.ls.fi.upm.es/udis/miembros/xavier/usabilityframework/>

## Appendix: Framework View by Development Activities

Table 1 shows the development activities view of the proposed framework.

**Table 1.** Framework for Usability Integration into the Software Process: View by Activity

Kind of Activity	Technique	UP	TN	GA	PSE	UIE	Rep	Stages			Basic Reference	
								IC	CC	EC		
Requirements Elicitation and Analysis (Reqs.)	Card Sorting	Y	L	H	M	H	3	N	N	N	Robertson, 01	
	Competitive Analysis	N	M	H	M	H	1	SW	N	N	Nielsen, 93	
	Affinity Diagrams	Y	L	H	M	H	1	SW	NU	NU	Beyer, 98	
	Contextual Inquiry	Y	H	M	M	H	3	SW	NU	NU	Beyer, 98	
	JEM (Joint Essential Modeling)	Y	M	M	H	M	1	N	N	N	Constantine, 99	
	Ethnographical Observation	N	H	M	M	M	2	SW	NU	NU	Preece, 94	
	User Analysis <sup>1</sup>	Personas	N	M	M	M	H	3	SW	NU	NU	Cooper, 03
		Structured User Role Model	N	L	M	H	M	1	SW	N	N	Constantine, 99
		User Profiles	N	H	H	H	H	5	SW	N	N	Mayhew, 99
	Task Analysis <sup>1</sup>	Essential Use Cases	N	M	H	H	H	1	N	N	N	Constantine, 99
		Task Scenarios	Y	M	M	M	H	1	N	N	N	Mayhew, 99
		HTA (Hierarchical Task Analysis)	N	M	M	H	M	1	SW	NU	NU	Preece, 94
	Develop Product Concept <sup>1</sup>	Scenarios and storyboards	Y	M	M	L	H	3	SW	NU	NU	Carroll, 97a
		Visual Brainstorming	Y	L	H	L	H	1	SW	NU	NU	Preece, 94
	Prototyping <sup>1</sup>	Paper Prototyping	Y	L	H	H	H	3	SW	N	N	Mayhew, 99
Requirements Specification (Reqs.)	Usability Specifications	N	M	M	M	H	4	N	N	N	Hix, 93	
Requirements Validation (Reqs.)	Inspections	N	M	H	M	H	4	N	N	N	Nielsen, 94	
	Heuristic Evaluation	N	H	H	L	H	6	N	N	N	Nielsen, 93	
	Collaborative Inspections	Y	M	M	M	M	1	SW	N	N	Constantine, 99	
	Cognitive Walkthrough	N	H	M	M	M	4	N	N	N	Carroll, 97b	
	Pluralistic Walkthrough	Y	L	M	M	M	4	SW	N	N	Bias, 94	

<sup>1</sup> These are not kinds of SE activities, but HCI activities included in this view to offer the framework user a structured vision of the 15 HCI techniques which may be applied in Requirements Elicitation and Analysis activities

Kind of Activity	Technique	UP	TN	GA	PSE	UIE	Rep	Stages			Basic Reference	
								IC	CC	EC		
Interaction Design (Design)	Menu-Selection Trees	N	L	M	H	H	1	N	N	N	Shneiderman, 98	
	Interface State Transition Diagrams	N	L	H	H	M	2	N	N	N	Hix, 93	
	Product Style Guide	N	H	M	M	M	1	NU	SW	N	Mayhew, 99	
	Navigation Maps	N	M	H	H	M	1	N	N	N	Constantine, 99	
	Interface Content Model	N	M	H	M	M	1	N	SW	N	Constantine, 99	
Design	Impact Analysis	N	M	M	H	M	3	NU	N	N	Hix, 93	
	Organizing Help by Use Cases	N	M	M	H	M	1	N	N	N	Constantine, 99	
Usability Evaluation (Evaluation)	Expert Evaluation	Inspections	N	M	H	M	H	4	N	N	N	Nielsen, 94
		Heuristic Evaluation	N	H	H	L	H	6	N	N	N	Nielsen, 93
		Collaborative Inspections	Y	M	M	M	M	1	SW	N	N	Constantine, 99
		Cognitive Walkthrough	N	H	M	M	M	4	N	N	N	Carroll, 97b
		Pluralistic Walkthrough	Y	L	M	M	M	4	SW	N	N	Bias, 94
	Usability Testing	Thinking Aloud	Y	M	H	L	H	5	NU	N	N	Nielsen, 93
		Post-Test Information	Y	M	H	M	H	1	NU	N	N	Constantine, 99
		Performance Measurement	Y	M	M	M	M	3	NU	N	N	Dumas, 99
		Laboratory Usability Testing	Y	M	M	M	M	4	NU	N	N	Hix, 93
	Follow-Up Studies of Installed Systems	User Feedback	Y	L	H	H	H	3	NU	NU	SW	Nielsen, 93
		Questionnaires, Interviews and Surveys	Y	M	H	M	M	3	NU	N	SW	Mayhew, 99
		Logging Actual Use	N	H	M	H	M	5	NU	N	SW	Shneiderman, 98

In order to use this view of the framework, a developer must look for the kind of activity he or she is interested in, and choose one or more of the proposed HCI techniques, according to the characteristics the organization values most (UP: User Participation, TN: Training Needs, GA: General Applicability, PSE: Proximity to SE, UIE: Usability Improvement/Effort Ratio, Rep: Representativeness), and to the stage where their application is intended (IC: Initial Cycles, CC: Central Cycles, EC: Evolution Cycles). Techniques labeled as "Very useful" are represented on a white background, while techniques labeled as "Useful" are represented on a gray background. The values for technique characterization are represented as follows: Y: Yes, N: No, H: High, M: Medium, L: Low. The values for development stage suitability are as follows: SW: Specially Well-matched, N: Neutral, NU: Not Usual.