Improving Software Engineering Practice with HCI Aspects

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. Techniques from the HCI (Human-Computer Interaction) field have been used for the development of usable software products for a long time, but their use is often not integrated with software engineering practices. In this work we describe an approach for bridging the gap between software engineering and HCI, by offering orientation to software practitioners on the application of HCI techniques and activities. For this purpose, we have carried out a survey in HCI literature to define the activities in a user-centered development process, and to select the HCI techniques that are more appropriate for integration into software engineering practice.

1 Introduction

According to ISO Standard 9241-Part 11, usability is "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" [7]. Usability is a critical quality factor for the success of a software product. In this direction, Larman states that there is probably no other technique with greater disproportion between its importance for the success of software development and the lack of attention and a formal education, as usability engineering and the design of the user interface [9]. Nevertheless, a change can be seen in the attention paid to usability. An increasing number of software development companies are beginning to consider usability as strategic for their business, and they are pursuing the aim of integrating HCI practices into their software engineering processes. Some proposals for integration ([1], [14]) present ad-hoc solutions which have been created for particular software development organizations, but they lack a generic approach to be applied to organizations with different characteristics.

Techniques aimed to increase the usability level of the software product are applied following development processes which are particular to the HCI field, and these processes are not formalized from the point of view of software engineering. Therefore, they are not easy to transfer to the formalized software engineering processes. One of the virtues of the HCI field lies in its interdisciplinary nature, but this characteristic is at the same time the greatest obstacle for its integration with software engineering. While the theoretical and practical basis of HCI comes from Sociology, Psychology, Ergonomics, Graphical Design and so on; software engineers have a clear engineering focus. Both fields speak different languages, and they approach software development from a very distinct perspective.

C.V. Ramamoorthy, R.Y. Lee, and K.W. Lee (Eds.): SERA 2003, LNCS 3026, pp. 349–363, 2004.

According to Constantine and Lockwood [2], the classic view of software quality has focused primarily on internal efficiency and reliability of the code in operation; but the view of software begins to shift outward from a limited internal perspective to an external one that more fully considers customers and end users. Usability, along with utility and capability, are clearly seen as key factors in software quality. Software development is mostly focused on internals, in processing logic and data organization to fulfill narrow concepts of functional objectives. These aspects of the software system are almost completely alien to the final user of the product. However, the system-user interaction has been traditionally considered as a secondary issue. Despite mentioning as an objective to build a system that satisfies user needs, after establishing a set of requirements, the development effort is mostly carried out without further contact with the final users (at least not before a first "polished" version of the software product is produced). Usability is often identified with just the design of the graphical user interface, a part of the system that is developed at the end of the software development process. This kind of approach is responsible for the development of systems with a very low usability level, whose usability problems are identified once their correction is too costly.

On the contrary, HCI experts study the users and the way they interact with the system from the beginning of the development effort. HCI experts employ a set of techniques for interaction design, and for evaluation of software products with real users.

The aim of the present work is to approach the integration of HCI techniques into the software development process from a software engineering perspective, making possible the application of HCI techniques by software developers (not HCI experts), or at least that software developers may incorporate the 'caring for usability' development philosophy present in HCI practices into their development practices. For this purpose, we have begun by studying the characteristics of a user-centered development process (the approach to development taken in HCI) in the HCI literature. Then we have identified the activities in a user-centered development process, and we have finished the HCI literature survey by studying the classification of the main HCI techniques according to a user-centered development scheme. In order to make this scheme understandable by software developers, we have finally mapped the activities in a user-centered process to the usual activities undertaken in a generic software development process, as understood in software engineering. Average developers may use our proposal to decide how and where the HCI techniques may fit with the rest of techniques they usually employ. Our proposal is generally applicable to interactive software development, given that the existing development process is based in iterative development.

The research presented in this work has been carried out as part of the project STATUS, financed by the European Commission (IST - 2001 - 32298). The project includes between its objectives, the output of methodological guidelines for the integration of usability techniques into the software process, which we are presenting here.

2 Definition of a User-Centered Development Process

As a first step for the integration of HCI techniques and activities into the software development process, we have carried out an HCI literature survey to identify the

characteristics that a software development process should have for it to be considered user-centered and, therefore, support the development of a final product with a high level of usability. These characteristics may be used by any organization to decide whether its software process can serve as a basis for the integration of usability techniques into software development or, on the contrary, it has to consider migrating to another type of process if it really intends to go for usability.

Albeit strictly in reference to user-centered *design*, Preece et al. gives a definition of user-centered that is potentially of interest for our process requirements search. It should [12]:

- be user-centered and involve users as much as possible so that they can influence the design,
- integrate knowledge and expertise from the different disciplines that contribute to HCI design,
- be highly iterative so that testing can be done to check that the design does indeed meet user requirements.

The ISO Standard 13407 on Human-Centered Design Processes for Interaction Systems [8] defines that the incorporation of a human-centered approach is characterized by the following:

- the active involvement of users and a clear understanding of user and task requirements;
- an appropriate allocation of function between users and technology;
- the iteration of design solutions;
- multi-disciplinary design.

Looking for a different point of view, we find that [2] defines the elements of a usage-centered approach as follows:

- Pragmatic design guidelines.
- Model-driven design process.
- Organized development activities.
- Iterative improvement.
- Measures of quality.

Shneiderman states that a process that supports usability needs to be non-hierarchical in the sense that it is neither strictly top-down nor bottom-up; and it is radically transformational, implying the production of interim solutions that could ultimately play no role in the final design [13]. This leads to an iterative development approach.

From the characteristics of a proper user-centered process detailed above, we can extract three main issues that need to be dealt with: user involvement; adequate understanding of user and task requirements; and iterative development. For the fulfillment of the first two requirements, we detail below a set of HCI techniques that may help to achieve them. The techniques specify when and how the user should be incorporated and what usability knowledge should be applied and when. On the other hand, iterative development is an intrinsic development process requirement. Therefore, according to [12], the organization's design process should be highly iterative to support usability and, consequently, to be able to incorporate the HCI practices.

Iterative development is a must. The usability level of the system cannot be predicted in advance. Some kind of usability evaluation is needed at the end of every iterative cycle. Therefore, the requirement of an iterative process is closely linked to the need to perform quality measures at the end of each cycle, and it is the only requirement to be met by a development process applied by a software development organization, for it be a candidate for the integration of HCI aspects.

The other two requirements: user involvement and adequate understanding of user and his or her tasks, are also changes in developers' way of doing things, although these changes can be accomplished by the application of HCI techniques.

3 Representative HCI Activities

The HCI field is diverse, and there is no general agreement on the set of activities that are part of a user-centered development process. For that reason, we have performed a literature survey in order to obtain a set of activities/tasks that lead to the development of software systems with an acceptable usability level.

Although software engineering has made efforts towards software process formalization, HCI authors have not strived for formality. On the contrary, they propose tasks, activities, process heuristics and advice, which are not integrated into a process that can be used as a framework for development. The sources vary as to the extent of formalization. The set of usability-related activities proposed in the HCI field are detailed in Table 1, where sources follow an order of increasing formalization from left to right. We have analyzed the activities proposed by the different authors in order to extract the common ones or, at least, the activities that are at the same abstraction level and are common to several sources. Our aim is to be able to easily compare the different proposals, and, therefore, we have grouped activities that refer to the same concept in the same row. Each row has been labeled (first column in the table) with the most general term or the term more often used by the authors studied, and there is one column per author that contains the respective activity that they propose. Where the author packs several tasks into the same activity, the complete name given for the activity (for example, Systems/ Tasks / Functional / User Analysis) has been included in the table. Some authors describe a generic activity that includes the activity we are considering as a subtask. In these cases, the specific subtask is highlighted in italics. On the other hand, where the author proposes several activities that match one of our activities, they are listed using an asterisk (*). For activities not mentioned in the source, the cell contains a dash ('-').

There is a clear trend in most of the sources considered as regards the activities to be done: usability specifications, prototyping and usability evaluation. The specification of the context of use, either as a complete analysis of a variety of user and organizational issues or just with an aim of knowing the user, is also quite prevalent among the different authors. We have found more discrepancies, and less information on other design activities, like the development of the product concept and the interaction design. While some authors give no clues as to the design activity, apart from labeling it as user-centered or advocating iterative design, Constantine and Lockwood [2] are more specific with respect design issues, criticizing the trend in usability engineering that focuses almost exclusively on usability testing.

Table 1. Usability Activities by Source.

Activity	Nielsen93	ISO99	Shneiderman 98	Hix93	Preece94	Wixon 97	Constantine 99
SPECIFICATION OF THE CONTEXT OF USE	Know the user	Understand and specify the context of use	Perform research and needs analysis	Systems/ tasks / functional / user analysis	Task analysis / functional analysis	* Specify and categorize the users * Conduct a task analysis	Task modeling
USABILITY SPECIFICATIONS	Goal Settings	Specify the user and organizational requirements	Design concepts and key-screen prototype (*create specific usability objectives based on user needs*)	Requirements / Usability Specifications	Requirements specification	* Define quantitative usability goals * Set levels of desired usability for each goal	-
DEVELOP PRODUCT CONCEPT	-	-	Develop product concept	Conceptual design	Conceptual design / formal design	-	-
PROTOTYPING	Prototyping	Produce design solutions (*make design solutions more concrete using simulations, models, mock-ups, etc.*)	Design concepts and key-screen prototype	Rapid prototyping	Prototyping	-	-
INTERACTION DESIGN	Iterative Design	Produce design solutions	Do iterative design and refinement	Design & design representation	Conceptual design / formal design	-	Interface content modeling
USABILITY EVALUATION	Interface Evaluation	Evaluate design against requirements	Do iterative design and refinement (*conduct full-scale usability tests*)	Usability evaluation	Evaluation	Test the product against usability goals	Usability inspection

The resulting usability activities (the left column in Table 1) are represented in Fig. 1, grouped according to the generic kind of activity to which they belong: Analysis, design or evaluation. Note that the Specification of the Context of Use is decomposed in User and Task Analysis. Some authors ([5], [15]) differentiate between both activities, even if they recognize that they are closely related. We have chosen the terminology of the Standard ISO-13407 [8] because it better reflects the close relationship between both subactivities.

Prototypes are widely used in other fields than HCI, in particular related to iterative development, but what HCI may offer is the particular usage of prototyping in order

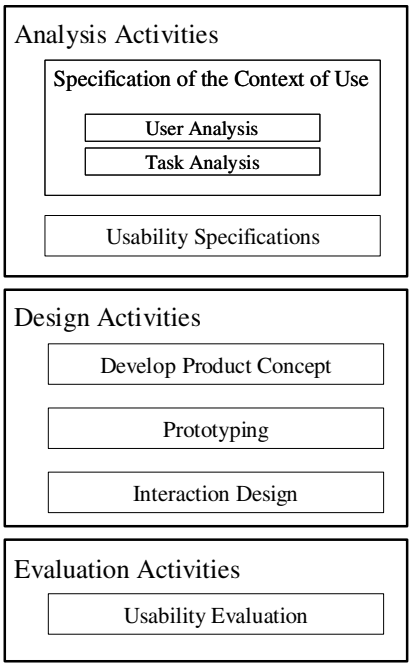

Fig. 1. Activities in a User-Centered Development Process.

to get greater degrees of user involvement, and to consider alternative designs. The most useful prototypes for this purpose are the less elaborate ones, such as paper prototypes.

Apart from Prototyping, the other two design activities identified have no such wide presence in HCI literature. The development of the product concept is based on mental models ([11], [12]): When the product concept is vague, ambiguous, inconsistent or obscure, there will be a divergence between the user mental model of the system and the design model that developers work with. The activity of Develop the Product Concept is not alien at all to software engineering; it is a modeling effort aimed to ensure a proper communication between the members of the development team. The pursuit of consistence and logic in a design is a clear engineering endeavor, and we just need to add the finality of producing a product concept that matches the expectations and needs of the final user.

Interaction Design is an activity that is not defined in great detail, and its definition noticeably varies between different authors. Interaction design and the design of the user interface are closely related, but they are different [3]. Some authors refer to "User Interface Design" ([10], [5], [13]), while others use the term "Interaction Design" ([12]) or just "Design" ([15], [8]). Additionally, Constantine and Lockwood refer to "Dialogue Design" or "Visual Design" [2]. As the design of the interaction is critical for the usability level of the final product, we have incorporated Interaction Design as an activity in the set of activities in a user-centered development process in Fig. 1.

Table 2. Analysis-Related Techniques.

Analysis-Related Techniques				Nielsen93	Preece94	Hix93	Shneiderman98	Constantine99
Specification of the Context of Use	Functional Analysis (5)			Functional Analysis		Functional Analysis		
	Needs Analysis (5) (8)					NeedsAnalysis		
	Competitive Analysis			Competitive Analysis				
	Financial Impact Analysis (7) (8)			Financial Impact Analysis				
	Contextual Inquiry				Contextual Inquiry	Contextual Inquiry		
	Ethnographic Observation				Ethnography		Ethnographic Observation	
	Sociotechnical Approach (1) (8)				Sociotechnical Approach			
	User Analysis	Structured User Role Model						Structured User Role Model
		User Profiles (4)		Individual User Characteristics		User Profiles	Usage Profiles	
		Operational Modeling						Operational Modeling
	Task Analysis	Essential Use Cases						Essential Use Cases
		HTA (4)			HTA			
		Cognitive Task Analysis			Cognitive Task Analysis			
			GOMS	GOMS	GOMS		GOMS	
			TAG (4)	TAG			TAG	
			Object-action Interface Model (4)				Object-action Interface Model	
		Scenarios		Scenarios			Scenario Development	
Usability Specifications	Based on Benchmark Tasks				Benchmark Tasks	Benchmark Tasks		
	Based on Preference Questionnaires					User Questionnaires		

4 Selection of HCI Techniques

After obtaining a characteristic set of HCI activities, we have continued our literature survey by focusing on the set of techniques commonly applied in the HCI field. The same sources as for the HCI activities survey were chosen, except for [15] and [8]. We have not considered these two sources, because they focus on activities in a usability-oriented process and give few details on the individual techniques to be used.

Table 3. Design-Related Techniques.

Design-Related Techniques				Nielsen93	Preece94	Hix93	Shneiderman98	Constantine99
Develop Product Concept	Conceptual Design					Conceptual Design		
	Post-It Notes							Post-It Notes
	JEM							JEM
	Visual Brainstorming				Visual Brainstorming			
Prototyping	Prototyping Strategies (6)	Rapid Prototyping			Rapid Prototyping	Rapid Prototyping		
		Incremental Prototyping			Incremental Prototyping			
		Evolutionary Prototyping			Evolutionary Prototyping			
	Kinds of Prototypes	Requirements Animation	Mock-ups (Limited Impl.)					Active Prototypes
		Non-functioning Prototypes	Chauffeured Prototypes		Chauffeured Prototyping			
			Paper Prototypes					Passive Prototypes
			Wizard of Oz		Wizard of Oz			
Interaction Design	Screen Pictures					Screen Pictures		
	Use Cases							Use Cases
	Grammars (1)						Grammars	
	Menu-Selection and Dialog Box Trees						Menu-Selection and Dialog Box Trees	
	Context Navigation Map					Interface State Transition Diagrams	Transition Diagrams and State-charts	Context Navigation Map
	UAN (4)					UAN	UAN	
Other Design Techniques	Design Alternatives Management	*Both-And Design*						*Both-And Design*
		Parallel Design		*Parallel Design*				
		Impact Analysis		Impact Analysis	Impact Analysis	Cost / Importance Analysis		
	Help Design	Organizing Help by Use Cases						Organizing Help by Use Cases
	Design Rationale (5)	IBIS, PHI, Design Space Analysis, Claims Analysis			IBIS, PHI, Design Space Analysis, Claims Analysis			

Table 4. Usability Evaluation Techniques for Usability Testing.

Usability Testing Techniques				**Nielsen93**	**Preece94**	**Hix93**	**Shneiderman98**	**Constantine 99**
Usability Tests	Thinking Aloud			Thinking Aloud	Think Aloud Protocol	Concurrent Verbal Protocol Taking		Talk to Me (think out loud)
		Constructive Interaction		Constructive Interaction				
		Retrospective Testing		Retrospective Testing	Post-Event Protocol	Retrospective Verbal Protocol Taking		
		Critical Incident Taking				Critical Incident Taking		
		Coaching Method		Coaching Method				
	Measured Performance							Measured Performance
	Post-Test Feedback							Post-Test Feedback
	Laboratory Usability Testing						*Usability Testing and Laboratories*	*Laboratory Usability Testing*
	Field Usability Testing	Direct Observation		Usability Assessment through Observation	Direct Observation			
		Beta-Testing						Beta-Testing
		Indirect Observation	*Video Recording*		*Video Recording*	*Video-taping*		
			Verbal Protocol		*Verbal Protocol*	*Audio-taping*		

We have experienced the same difficulty than in the previous section. The HCI field is very heterogeneous, and we have found a great diversity of techniques. After merging the techniques suggested by different authors that refer to the same basic technique, we still have eighty-two techniques. This is an excessive number of techniques to provide to software developers, especially given that some of them are redundant and a certain number have almost no interest for general software development projects. For the purpose of selecting the techniques more appropriate for integration into software engineering practice, the techniques have been divided into groups, according to a classification of the main kind of activity. We have represented the techniques in several tables: One table for analysis-related techniques (Table 2),

one table for design-related techniques (Table 3), and four tables for evaluation-related techniques (Tables 4 to 7). Each table contains a column for each author, and the left-hand columns specify the technique category and chosen name. For each technique, we have chosen the name we consider to be the most representative. Techniques that are in the same row refer to the same basic technique. The possibility of variants of some techniques is also considered; and then the general technique and each of the variants have their corresponding rows (for example, Table 4 shows the Thinking Aloud technique along with four of its variants). The tables also show the selected as candidate techniques. Techniques on a white background are selected as candidates for inclusion in the software development process. Techniques that have not been selected appear on a grey background, and they have between brackets an indicator of the reason for not being selected, as detailed in Table 8. A more detailed description of the selection process can be found in [4].

A technique is discarded due to one or more of the reasons in Table 8. The techniques appearing in italics have been selected, albeit for optional application when the project meets certain characteristics. Analysis techniques are summarized in Table 2, and design techniques in Table 3.

Evaluation techniques are summarized in four tables: Table 4 presents techniques for usability testing, Table 5 for expert reviews, Table 6 for follow-up studies of installed systems and, finally, Table 7 summarizes the rest of usability evaluation techniques.

Table 5. Usability Evaluation Techniques for Expert Reviews.

Expert Review Techniques			**Nielsen93**	**Preece94**	**Hix93**	**Shneiderman98**	**Constantine99**
Expert Reviews	Heuristic Evaluation		Heuristic Evaluation	Heuristic Evaluation	Heuristic Evaluation	Heuristic Evaluation	Heuristic Evaluation
	Inspections	Conformance Inspections		Standards Inspection		Guidelines Review	Conformance Inspections
		Consistency Inspection		Consistency Inspection		Consistency Inspection	Consistency Inspection
		Collaborative Usability Inspections					Collaborative Usability Inspections
	Walkthroughs	Pluralistic Walkthrough	Pluralistic Walkthrough	Pluralistic Walkthrough			Pluralistic Usability Walkthrough
		Cognitive Walkthrough		Cognitive Walkthrough		Cognitive Walkthrough	Cognitive Walkthrough

Table 6. Usability Evaluation Techniques for Follow-up Studies of Installed Systems.

Techniques for Follow-Up Studies			**Nielsen93**	**Preece94**	**Hix93**	**Shneiderman98**	**Constantine99**
Follow-up Studies of Installed Systems	Questionnaires		Questionnaires and Interviews	Questionnaires and Surveys			
	Interviews		Questionnaires and Interviews	Interviews		Interviews and Focus Group Discussions	
		Structured Interviews		Structured Interviews	Structured Interviews		
		Flexible Interviews		Flexible Interviews			
	Focus Groups		Focus Groups			Interviews and Focus Group Discussions	
	Logging Actual Use		Logging Actual Use	Software Logging	Internal Instrumentation of the Interface	Continuous User-Performance Data Logging	
		Time-Stamped Key presses		Time-Stamped Key presses			
		Interaction Logging		Interaction Logging			
	User Feedback		User Feedback			Online Suggestion Box or Trouble Reporting	
		Online or Telephone Consultants				Online or Telephone Consultants	
		Online Bulletin Board or Newsgroups				Online Bulletin Board or Newsgroups	
		User Newsletters and Conferences				User Newsletters and Conferences	
	Surveys			Questionnaires and Surveys		Surveys	

Table 7. Other Usability Evaluation Techniques.

Other Usability Evaluation Techniques	Nielsen93	Preece94	Hix93	Shneiderman98	Constantine99
Experimental Tests (2)		Traditional Experiments		Controlled Psychologically Oriented Experiments	
Predictive Metrics		Analytic Evaluation Methods			Usability Assessment Based on Predictive Metrics
Acceptance Tests (5)				Acceptance Tests	
Cooperative Evaluation (2) (8)		Cooperative Evaluation			
Participative Evaluation (1)		Participative Evaluation			

Table 8. Reasons for Discarding a Technique.

1	It is a special technique for projects with specific characteristics, so it is not generally applicable
2	It is alien to software engineering, so developers will find it very difficult to learn to use it
3	Its application will require the use of extra resources from outside the project team
4	It is made redundant by another selected technique. That is, the expected benefits provided by the application of the technique are already covered by a selected technique, and the selected technique offers some additional advantages
5	It is not specifically an HCI technique, so it does not make sense to include it in an HCI addition to the development process
6	It deals with development process issues, and there are other reasons apart from usability to be taken into account. It must be dealt with in the context of the whole development process
7	The technique is directed at gaining management support for usability activities in the development process. We are working with the hypothesis of an organization that is already willing to incorporate HCI practices, so this kind of support is pre-requisite for the usage of our proposal
8	It is presented by just one author, and we consider that it is not generally accepted as a usability technique in the field. This reason will be considered only in conjunction with other reasons, never by itself

5 Fitting HCI Activities into Mainstream Development

If we want that software developers use the HCI techniques that we have compiled, we need them to be expressed according to terminology and concepts that are familiar to developers. The scheme of activities and techniques we have obtained from the HCI field is not very useful for an average developer, because it is based on HCI concepts and terminology, which are somehow alien to them. Therefore, we need to adapt the activity scheme (and also the techniques) from HCI to the activities of a generic software development process.

For the definition of the set of activities in a generic development process, we have mainly based on the SWEBOK (SoftWare Engineering Body Of Knowledge) [6] developed by the IEEE Computing Society.

Table 9 shows the relationship between the activities in a user-centered development process and the activities in a generic development process.

Table 9. Mapping between Development Activities Affected by Usability and HCI Activities.

Development Activities Affected by Usability			**Activities in a User-Centered Development Process**
Analysis (Requirements Engineering)	Req. Elicitation		Specification of the Context of Use - User Analysis
	Req. Analysis	Develop Product Concept	Develop Product Concept
		Problem Understanding	Specification of the Context of Use - Task Analysis
			Prototyping
		Modeling for Specification of the Context of Use	Specification of the Context of Use - User Analysis
	Req. Specification		Usability Specifications
	Req. Validation		Walkthroughs (Usability Evaluation - Expert Evaluation)
Design	Interaction Design		Interaction Design
Evaluation	Usability Evaluation	Usability Testing	Usability Evaluation - Usability Testing
		Expert Evaluation	Usability Evaluation - Expert Evaluation
		Follow-Up Studies of Installed Systems	Usability Evaluation - Follow-Up Studies of Installed Systems

Regarding analysis, usability activities are intermingled with other analysis activities, so we will integrate HCI activities in analysis with the activities in a generic development process (as indicated in the SWEBOK) with which they are more closely related. There are two activities considered as design activities in HCI, but considered as analysis in software engineering: Prototyping is traditionally used in software engineering for the task of Problem Understanding, while the Development of the Product Concept is a kind of design known as innovative design, which is usually undertaken as part of Requirements Engineering. The SWEBOK does not consider innovative design as part of software design, but as part of requirements analysis efforts. In addition to that, we have that Walkthroughs are a kind of usability evaluation that can be used for the validation of the products of analysis, and that is the reason why it is mentioned in the mapping in Table 9.

Unlike analysis, we have in design and evaluation that HCI activities are quite independent from the rest of development activities, so we have added new activities to accommodate them in the scheme. Regarding design, we have defined a new development activity called Interaction Design.

Usability evaluation is also performed independently from the rest of evaluation activities, and for that reason we have defined a Usability Evaluation activity in the generic scheme. Usability evaluation has a high level of complexity due to the diversity of existing HCI techniques for that purpose, so we have decomposed this activity into the three main kinds of usability evaluation activities: Usability Testing, Expert Evaluation, and Follow-Up Studies of Installed Systems.

6 Conclusions

We have presented a strategy for the introduction of HCI techniques and activities into mainstream software engineering practice, by presenting HCI techniques and activities organized according to a scheme that may be mapped to the activities being carried out in any software development organization.

The main reasons for the current not integration of both disciplines (HCI and software engineering) lie in their different terminology and approach to software development. We propose to use software engineering processes and terminology to characterize HCI aspects, so they can be assimilated by average software developers.

For a software development organization wanting to improve the usability of their software products, it is very appealing to have a set of HCI techniques to be incorporated to their current software development process, without need for them to change it for a completely new process. The existing in-house development process does not need to be abandoned, except in the case that it is not iterative. Given the current trend towards iterative development, we consider that this only requirement for the existing software development process is not too restrictive.

For the elaboration of our proposal, we have collaborated with the two software development companies that are part of the STATUS project consortium. They work in the e-commerce domain, where usability is specially critical. After finishing the work presented in this article, we have offered the resulting scheme to both companies and we have received an encouraging initial response from them. With the feedback they provide after using the scheme in two real projects, we plan to refine the scheme to expand it in the direction(s) where developers need more guidance.

References

1. Anderson, J., Fleek F., Garrity, K., Drake. F.: Integrating Usability Techniques into Software Development. IEEE Software. vol.18, no.1. (January/February 2001) 46-53
2. 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 (1999)
3. Ferre, X., Juristo, N., Windl, H., Constantine, L.: Usability Basics for Software Developers. IEEE Software. vol.18, no.1. (January/February 2001) 22-29
4. Ferre X., Juristo, N., Moreno. A.M.: STATUS Project. Deliverable D.5.1. Selection of the Software Process and the Usability Techniques for Consideration (2002) http://www.ls.fi.upm.es/status/results/deliverables.html

5. Hix, D., and Hartson, H.R.: Developing User Interfaces: Ensuring Usability Through Product and Process. John Wiley and Sons, New York NY (1993)
6. IEEE Software Engineering Coordinating Committee: Guide to the Software Engineering Body of Knowledge - Trial Version 1.00. IEEE Computer Society, Los Alamitos, California (May 2001)
7. ISO: ISO 9241-11. Ergonomic Requirements for Office Work with Visual Display Terminals. ISO, Geneva, Switzerland (1999)
8. ISO: International Standard: Human-Centered Design Processes for Interactive Systems, ISO Standard 13407: 1999. ISO, Geneva, Switzerland (1999)
9. Larman, C.: UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process (2nd. Edition). Prentice Hall PTR (2001)
10. Nielsen, J.: Usability Engineering. AP Professional, Boston, MA (1993)
11. Norman, D. A.: The Design of Everyday Things. Doubleday (1990)
12. Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey. T.: Human-Computer Interaction. Addison Wesley, Harlow, England (1994)
13. Shneiderman. B.: Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley, Reading, MA (1998)
14. Radle, K., Young, S.: Partnering Usability with Development: How Three Organizations Succeeded. IEEE Software. vol.18, no.1. (January/February 2001) 38-45
15. Wixon, D., Wilson. C.: The Usability Engineering Framework for Product Design and Evaluation in Helander. M.G. et al. (eds.): Handbook of Human-Computer Interaction. Elsevier North-Holland (1997) 653-688