Improving Software Engineering Practice with HCI Aspects

Xavier Ferre
Universidad Politecnica de Madrid
xavier@fi.upm.es

Ana M. Moreno
Universidad Politecnica de Madrid
ammoreno@fi.upm.es

Abstract

Techniques from HCI 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" [1]. According to Constantine, usability is a key factor in software quality [2]. Furthermore, 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 [3]. Nevertheless, a change can be seen in the attention paid to usability and the HCI (Human-Computer Interaction) practices that ensure the usability of the software product. An increasing number of software development companies are beginning to integrate HCI practices into their SE (Software Engineering) processes. Some proposals for integration ([4], [5]) 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 [6], and these processes are not formalized from the point of view of SE. Therefore, they are not easy to transfer to the formalized SE 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 SE. Both fields speak different languages, and they approach software development from a very distinct perspective.

Software developers often identify usability 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. Even if, perhaps, usability issues should be handled just by HCI experts, the reality is that average developers make the majority of the numerous decisions that determine the ultimate usability of a software product [2].

The aim of the present work is to approach the integration of HCI techniques into the software development process from a SE perspective, making possible the application of HCI techniques by software developers (without HCI expertise) with the adequate training, or at least increasing usability awareness in software development teams if such training is not readily available. For this purpose, we have started 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 classified the main HCI techniques according to such activities. 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. Average developers may use our proposal to decide how and where the HCI techniques may fit with the rest of techniques they usually employ.

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.

Most HCI authors ([7], [8], [6], [2], [9]) agree in considering iterative refinement a basic characteristic of a user-centered approach. The usability level of the system cannot be predicted in advance, therefore, some kind of usability evaluation is needed at the end of every iterative cycle. User involvement is also mentioned in several sources as part of a user-centered approach ([7], [6], [8]). The remaining common characteristic of the user-centered approach is the focus on the adequate understanding of the user and his or her tasks ([8], [6], [10]). Therefore, for a software development process to be considered user-centered, it must have these three characteristics: to be based on iterative refinement, involve users, and work for an adequate understanding of users and their tasks. Only the first requirement is an intrinsic characteristic of the software process, since the other two conditions may be met through the application of specific HCI techniques. Thus, we identify iterativeness as the only requirement for a software process for it to be a candidate for the integration of HCI aspects.

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. The sources vary as to the extent of formalization, but none reaches the level of formalization that can be found in SE processes. 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 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. 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: Specification of the context of use, usability specifications, prototyping and usability evaluation. We have found more difference between different authors views in some design-related activities, but we can still identify the activities of develop the product concept and interaction design.

Table 1 - Usability Activities by Source

Activity \ Source	[12]	[1]	[10]	[8]	[7]	[11]	[2]
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 categorise the users * Conduct a task analysis	Task modeling
USABILITY SPECIFICATIONS	Goal Settings	Specify the user and organizational requirements	Design concepts and key-screen prototype	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	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	Usability evaluation	Evaluation	Test the product against usability goals	Usability inspection

The resulting usability activities (the left column in Table 1) are represented in Figure 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, following the terminology of the Standard ISO-13407 [6].

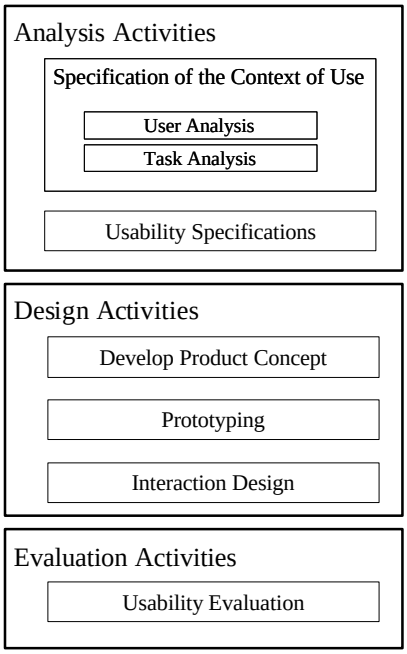

Figure 1 - Activities in a User-Centered Development Process

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 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.

The development of the product concept is based on mental models ([13], [7]), and it consists on elaborating a shared model of the product concept, making it explicit to avoid any divergence between the user mental model of the system, and the design model that developers work with. Regarding interaction design, it is not just about the visible part of the user interface, as interaction design is also about functionality, sequencing, content, and information access [8].

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.

We have experienced the same difficulty than in the previous section. The HCI field is 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 seventy-nine techniques. This is an excessive number of techniques to provide to software developers, specially given the fact that some of them are redundant, and also that a certain number of techniques belong to specific domains and have almost no interest for generic software development. Therefore, we have discarded certain techniques from our scheme, in order to reduce the complexity for the software developer. Table 2 details the criteria for discarding a technique.

Table 2 - 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 SE, 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.

We show in Table 3 and Table 4 the classification of HCI techniques related to analysis and design respectively. For reasons of space we are not detailing here the evaluation techniques. They represent the higher number of techniques, but they hold less interest for the integration with SE practices, since their application is quite independent from other testing activities in the overall development process.

Each table contains a column for each author, and the left-hand columns specify the HCI activity where they fit according to HCI literature. 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 tables also show the selected and discarded techniques. Techniques on a white background are selected as candidates for inclusion in the software development process. On the other hand,

discarded techniques appear on a grey background, and they have between brackets an indicator of the reason for not being selected, as detailed in Table 2. A technique is discarded due to one or more of the reasons in Table 2. The techniques appearing in italics have been selected, albeit for optional application when the project meets certain characteristics.

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. 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) [14]. Table 5 shows the relationship between the activities in a user-centered development process and the activities in a generic development process.

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 SE: Prototyping is traditionally used in SE 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 5.

Table 3 - Analysis-Related Techniques

Analysis-Related Techniques				**[12]**	**[7]**	**[8]**	**[10]**	**[2]**
Specification of the Context of Use	Functional Analysis (5)			Functional Analysis		Functional Analysis		
	Needs Analysis (5) (8)					Needs Analysis		
	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		

Table 4 - Design-Related Techniques

Design-Related Techniques				**[12]**	**[7]**	**[8]**	**[10]**	**[2]**
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 Implementation)					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 Statecharts	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			

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.

As mentioned above, 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. Due to the high number of usability evaluation techniques, 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 described a framework for the integration of HCI activities and techniques into the software development process. Our proposal is aimed to software development organizations wanting to introduce HCI aspects into their development practices.

HCI techniques and activities have been organized according to a scheme that may be mapped to the activities being carried out in any software development organization. This process is eased by the employment of SE concepts and terminology to characterize HCI aspects, so they can be assimilated by average software developers. Once this starting point is reached, the actual introduction of specific HCI techniques may be considered.

Iterative refinement is a must for any software development process where usability matters. Therefore, we can communicate to software development teams that an iterative process is the only one possible if usability is one of the software quality attributes that must be catered for. Given the current trend towards iterative development, we consider that this only constraint to the existing software development process is not too restrictive.

Table 5 - Mapping Development Activities Affected by Usability - 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

For the elaboration of our proposal, we have collaborated with two software development companies that 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 developers. 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 where developers need more guidance.

Acknowledgments

This research has been funded by the "Ministerio de Ciencia y Tecnología" of Spain, project number TIC2002-00320.

References

[1] ISO, "ISO 9241-11. Ergonomic Requirements for Office Work with Visual Display Terminals", ISO, 1999.
[2] L. Constantine, and L. Lockwood, "Software For Use: A Practical Guide to the Models and Methods of Usage-Centered Design", Addison-Wesley, 1999.
[3] C. Larman, "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process", 2nd Edition, Prentice Hall PTR, 2002.
[4] J. Anderson, F. Fleek, K. Garrity, and F. Drake, "Integrating Usability Techniques into Software Development", IEEE Software, vol. 18, no. 1, pp. 46-53, January/February 2001
[5] K. Radle, and S. Young, "Partnering Usability with Development: How Three Organizations Succeeded", IEEE Software, vol. 18, no. 1, pp. 38-45, January/February 2001.
[6] ISO, "ISO 13407.Human-Centred Design Processes for Interactive Systems", ISO, 1999.
[7] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey, "Human-Computer Interaction", Addison Wesley, 1994.
[8] D. Hix, and H.R. Hartson, "Developing User Interfaces: Ensuring Usability Through Product and Process", John Wiley and Sons, 1993.
[9] X. Ferre, N. Juristo, H. Windl, and L. Constantine, "Usability Basics for Software Developers", IEEE Software, vol. 18, no. 1, pp. 22-29, January/February 2001.
[10] B. Shneiderman, "Designing the User Interface: Strategies for Effective Human-Computer Interaction", 3rd Edition, Addison-Wesley, 1998.
[11] D. Wixon, and C. Wilson, "The Usability Engineering Framework for Product Design and Evaluation" in Handbook of Human-Computer Interaction, ed. by M.G. Helander et al. Elsevier North-Holland, 1997. pp 653-688.
[12] J. Nielsen, "Usability Engineering", AP Professional, 1993.
[13] D. A. Norman, "The Design of Everyday Things", Doubleday, 1990.
[14] IEEE Software Engineering Coordinating Committee, "Guide to the Software Engineering Body of Knowledge - Trial Version 1.00", IEEE, May 2001.