Human-Computer Interaction – INTERACT'03
M. Rauterberg et al. (Eds.)
Published by IOS Press, © IFIP, 2003, 3 pages (online version)

User Centred Design Process Model. Usability Engineering and Software Engineering Integration.

Toni Granollers

University of Lleida, E.U.P. Avda. Jaume II 69, Lleida, Catalonia - Spain

tonig@griho.net

Abstract: Usability is a fundamental attribute of today's interactive systems. The Human-Computer Interaction (HCI) discipline provides the foundations to develop usable applications but the development models used by the software industry in the development of applications are still those proposed by Software Engineering (SE). This paper summarizes my doctoral research by presenting a Process Model (PM) as a model for the development of interactive applications. The proposed model integrates the specific models and tasks of usability from the life cycle of SE with feedback from real projects. As a conclusion of the research, the idea of measuring the effort involved in usability when developing an interactive system is introduced.

Keywords: HCI, Usability Engineering, Software Engineering

1 Objective and significance of the proposal

In the design of a product, quality is one of the most important aspects when determining if one product is better than another; software industry is no exception to this factor. One of the most deterministic quality parameters of the interactive systems is usability (Bevan, 1999), and the most prestigious standardizations (IEEE 98, ISO 91) have been considering this factor for a long time.

The Human-Computer Interaction (HCI) discipline studies all the factors related to the interaction between the human and the computer with the objective to develop or improve the safety, utility, effectiveness and usability of interactive computer-based products. Existing textbooks provide sound coverage of HCI concepts and techniques but offer little guidance for comprehensive project activities.

Software Engineering (SE) is the computing discipline concerned with all aspects of software production (Sommerville, 2000) that provides a systematic approach to the analysis, design, implementation and maintenance of software.

A methodology that integrates HCI concepts and techniques with SE is still needed (Faulkner, 2000), although Usability Engineering (UE) emerged partly in response to this need. Digital Equipment Corporation professionals (Good, 1986) used this term to refer to the concepts and techniques used to plan, achieve and verify the system's usability objectives.

Today, in spite of this, most software project development teams carry out their work following one of the Process Models proposed by SE, which either does not consider usability at all (or allocates it to a very low level and too late on).

We think that this is because:

- The current UE models differ considerably from the SE models; so the developers look on this as a radical change and reject the unknown model.
- Existing usability models are too complex (Mayhew, 1999; Gerrit 1990; Brink, 2002; Rosson, 2002), mainly for those who are not computer specialists.
- Company managers don't believe that usability is cost justified (Bias & Mayhew, 1991). They see a larger development process without a direct sales increment.
- Marketing directors sell the image by justifying an easy-to-use factor from subjective appreciations.
- Project managers, designers and developers see HCI as an academic subject and not useful for real projects.

All these considerations motivated us to propose our own Usability Engineering Process Model as a bridge between the two disciplines, and this is

discussed in the following sections.

2 The Usability Engineering Process Model (UEPM)

An important problem in developing interactive systems is that development teams are multidisciplinary, which means that a great variety of people will be working with the same goal but with different mental models, and therefore the method must be comprehensible by all of them and not only by the software engineers.

Today, as it has been mentioned, software developers have a mental model based on SE principles and our proposal extends this aspect instead of changing it.

2.1 The Process Model outline

Figure 1 illustrates a diagram of the Process Model (PM). Since explaining and detailing the activities concerned with each node or phase of this diagram is not the objective of this paper, only the main features will be explained.

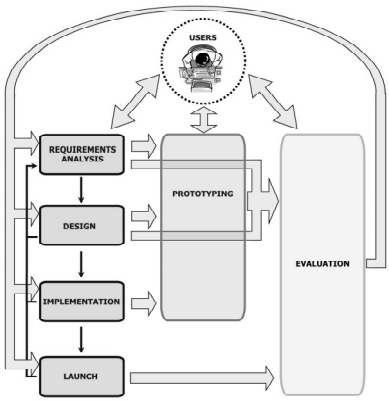

Figure 1: Usability Engineering Process Model.

1. *Conceptual Organization.* All the HCI knowledge is fitted into a specific block and the development team members always know which activity is being developed and what the related activities are.
2. *Three Basic columns.* The PM clearly defines three visible columns as a basis:
   - SE on the left
   - Prototyping, in the centre, grouping all the techniques that facilitate the next column
   - Evaluation, on the right, uniting all the validation methods
3. *The User.* A User Centred Process Model has the user as the most important part. This scheme reflects this meaning at the first glance.
4. *Iterativity*. Creating software, like any engineering process, has a more or less important phase based on iterations of some steps focused on obtaining the right solution. The arrows of the model specify this iterativity concept. Some correspond to the SE and the rest integrate this cycle to accomplish the usability parameters.

The model gives *freedom to the development team* without indicating a fixed path to follow. The team-members' experience using the requirements analysis and the evaluations' results will determine how many iterations have to be done, and what the most appropriate techniques are to use in each iteration.

2.2 Application

The UEPM on the one hand has its foundations in: (1) HCI discipline, which contributes –among others– with all the techniques and experiences known for design of interfaces focused on their users, and, (2) SE, which contributes with an extensive range of methodologies (widely used and tested).

Our intention is to provide a working tool to help development teams methodologically. It specifies neither the use of a certain programming language, nor any specific technology, nor any factor that can determine the application, but just the opposite; it is device and technology independent. It can be applied to developing traditional desktop applications and web-based applications. Moreover it is applicable to any interaction paradigm (desktop, ubiquitous, virtual reality and augmented reality).

Web development, just as an example, is defined by new rules; nevertheless a lack of methodological standards and conventions makes *development* and deployment *difficult* and *complex*. In our experience, UEPM solves this problem.

2.3 SE protection activities

Guaranteeing that all the activities carried out at every stage of the process are done with total fidelity to the UEPM is not an easy task. For this reason, we use some *protection activities* provided by SE, which give support to the process of development with the main purpose of obtaining a product of demonstrable quality (Pressman, 2001). It provides a way of integrating SE with usability practices in our model.

These protection activities used are: (i) *Configuration Management* (CM) as the activity that allows "one to manage the change" throughout all the software's service life; (ii) *Software Quality Assurance* (SQA) that on the basis of designing specific plans that consist of inspection, revisions

and tests made throughout the service life, allows one to assure that each product fulfils the requirements for which it is intended; and, (iii) *Risk Management*, which is a software engineering practice with processes, methods, and tools for managing risks in a project.

All the above mentioned activities together with an accurate planning surround the UEPM and give consistence to the Process Model.

3 Results and Goals to achieve

Results. We are developing different (web-based, desktop-based, ubiquitous-based) applications which are being used to test the validity of the presented methodology. The final evaluations of these applications have yielded results with an elevated level of usability without further increases in development time.

The most relevant projects where the UEPM has been applied are: implementation of a *ubiquitous reception-hall for a real industry* (Granollers, 2003) with an uninitiated user, first phase of *the use of Augmented Reality during a visit to an archaeological site* (Granollers, 2002) and *two websites* (one focused on children). Moreover, our methodology has been incorporated in an official national research project proposed to the Spanish government where three university research groups, with more than 30 qualified investigators, will work together. This project is titled as "*ITACA: A methodology (and tools) based on cognitive models to develop interactive and collaborative systems*".

Goals to achieve. Research on usability will never be complete, because there is no such thing as a perfect user interface (Nielsen, 2003). In spite of this, we still need to have "*something*" that indicates how usable the developed system is.

Our first attempt was to define and solve a so-called Usability Quality Index to indicate a figure for the usability level of the relevant application. However, to measure something as subjective as usability is virtually impossible.

Therefore, we changed the direction of our investigation to obtaining an indicative value for the amount of work involved in achieving system usability. So, we define *Usability Effort* (UE) as the *measurement that indicates the weighted resources and tasks employed during the development of an interactive application to get a good usability level*. The idea comes from the fact that, in theory, if someone made a greater effort in doing something, the result should be better than that achieved by another with less effort.

For this**,** we defined and use a *Configuration Management Work-Sheet* that chronologically reflects all the activities done indicating the corresponding UEPM phase.

At the time of writing this paper we still have no successful mathematical function that will enable us to determine this Usability Effort.

We emphasise this aspect heavily because we believe that knowing this value will increase the quality of future interactive systems.

References

Bevan N. (1999), Quality in Use: Meeting User Needs for Quality, Journal of System and Software.

Bias R.G., Mayhew D.J. (1991), Cost-justifying usability. IEEE Software.

Brink, T., Gergle D., Wood, S.D. (2002). Design web sites that work: Usability for the Web: Morgan-Kaufman

Faulkner X., Culwin F. (2000), Enter the Usability Engineer: Integrating HCI and Sw. Eng.. ACM

Gerrit, C. van der Veer (1990), *HCI: learning, individual differences, and design recommendations*. PhD.

Good M., et. al . (1986), User-derived impact analysis as a tool for usability engineering. Proceedings of Human Factors in Computing Systems: CHI'86.

Granollers, T et. al., (2002), Vilars. A new dialogue model applying Augmented Reality, Proc. of DSV-IS 02. Rostock (Germany).

Granollers, T. et.al. (2003), Developing an Ubiquitous reception-hall using the UCD Usability Engineering Process Model, Proceedings of HCI Intl.03. Crete

Mayhew, D.J. (1999), The Usability Engineering Lifecycle. Morgan-Kaufman.

Nielsen J. (2003), PR on Websites: Increasing Usability, March, 2003 Jakob Nielsen's Alertbox

Pressman, R. (2001), Software Engineering: A Practitioner's Approach. 5th Edition. Mc Graw-Hill

Rosson M.B., Carroll J.M. (2002), Usability Engineering: scenario-based development of HCI. Morgan-K

Sommerville, I. (2000), Software Engineering. 6th edition