INFORMATION SOCIETY TECHNOLOGIES

(IST) PROGRAMME

STATUS

"Software Architecture That Supports Usability"

W1: Project Management

D 1.8 Project Presentation

Version:	**1.1**
Submission Date:	**01/06/2002**
Authors:	**IHG**
Partners:	**IHG, UPM**

Stage:	**Confidentiality:**
[ ] Draft	**[x] Public - for public use**
[ ] To be reviewed by WP participants	**[ ] IST – for IST programme participants**
[ ] Pending of approval by next consortium meeting	**[ ] Restricted – for STATUS consortium and PO**
[x] Final / Released to CEC	

DOCUMENT CONTROL

Registration of Changes

Date	Version	Author	Comments
30.05.2002	Draft	IHG	First draft
31.05.2002	Draft	IHG	Changes and modifications
01.06.2002	1.1	UPM	Adjustments on cover page

Distribution

Recipient	Delivery Date

List of STATUS Related Documents

Document Name	Version

TABLE OF CONTENTS

1. PROJECT OBJECTIVES

The aim of the STATUS project is to study and determine the connections between software architecture and the usability of the resultant software system. The project will explain the characteristics of software architectures that improve software usability. Usability is a key quality attribute in software products. Usability benefits are patent from the viewpoint of both the user organization (for example, the benefits due to more efficient use of information technologies) and the developer organization (for example, the extent to which its applications are used).

A new line of software engineering research has recently (as of 1996) been opened that aims to discover how different quality attributes relate to software architectures. The traditional software quality attributes are reliability, efficiency, maintainability, However, there is no doubt (and many authors defend this stand) that usability is another software quality attribute. However, the relationships between software usability and software architecture have not yet been studied. The research conducted so far has focused on other quality attributes like performance or maintainability.

As there is a wide range of different architectural styles and the set of possible styles is still in the process of definition and standardization, this project will focus on e-commerce applications and their associated architectural styles. E-commerce applications have been chosen because they are of fundamental importance in the Information Society and because usability is a critical factor in these applications. This makes e-commerce applications a suitable benchmark for studying the relationships between and implications of architecture and usability.

1.1 Scientific and technological objectives

The research proposed in this project poses several scientific and technological objectives to achieve the above aim:

O1. **Identify usability attributes that are possibly affected by software architecture**. Researchers in the field of usability have already produced classifications of usability attributes. However, these attributes can be affected by other software and development characteristics apart from architecture. Therefore, the usability attributes need to be analyzed to identify which are affected by software architecture. Also, new usability attributes directly related to architecture could even be output and defined. The identified attributes should be able to be valued in order to examine whether they increase or decrease in relation to variations in the architecture.

O2. **Determine how usability attributes can be influenced by software architecture**. This objective involves studying the relationship between architecture and usability from several angles: evaluate how existing architectural styles support or are detrimental to usability attributes; define how some usability attributes can be translated into architectural details to be added to existing styles; study the suitability of defining a new architectural style that supports usability.

O3. **Identify architectural patterns that are repeated in the e-commerce domain and how they can be modified to improve the usability of the resultant software systems**. Most applications today (including e-commerce applications) do not respect a pure architectural style. On the contrary, they are usually a mixture of styles. Therefore, having studied the relationship between pure architectural styles and usability attributes, this study has to be instantiated to a particular type of applications. This project will address e-commerce applications.

The above objectives deal with usability at the architectural level. However, architecture can support, but it alone cannot guarantee that the final system will be highly usable. This project therefore poses a fourth objective, which moves away from architecture and focuses on another aspect of development that is critical for supporting final system usability: the development process. Here, the aim is to identify the parts of the development process in which usability has to be considered and how. This objective can be summarized as:

O4. **Modify traditional software development processes with techniques and activities that improve software usability**. One reason why software developers have not contemplated the advances made in the field of usability is because they have not been integrated with traditional development techniques and activities. So far, it has been left to each individual developer to decide when and how to apply an interesting technique from the usability field in development. This project aims to provide an integrated software process that incorporates techniques and activities proper to the field of usability with traditional software development techniques and activities. This integrated software process will help the developer to position the architectural results of this project within the development process framework.

1.2 Project Results

The materialization of these objectives will provide the following results:

R1. A description of the strengths and weaknesses of existing architectural styles with respect to usability

R2. Architectural model that supports usability

R3. Architectural model to support the usability quality attribute instantiated for the e-commerce domain

R4. A process model for the integration of usability techniques into the software process

R5. Real projects in the e-commerce domain demonstrating project results (R2, R3 and R4)

2. LIST OF PARTICIPANTS

CF 1: **Information Highway Group** (Financial Coordinator)
IHG (Spain) Start of project -End of project

CS 2: **Universidad Politécnica de Madrid** (Scientific Coordinator)
UPM (Spain) Start of project -End of project

P 3: **University of Groningen** (Principal Contractor)
RuG (Netherlands) Start of project -End of project

P 4: **Imperial College of Science, Technology and Medicine** (Principal Contractor)
ICSTM (United Kingdom) Start of project -End of project

P 5: **LOGICDIS S.A** (Principal Contractor)
LOGICDIS (Greece) Start of project -End of project

3. PROJECT WORKPLAN

3.1 Workpackages

The above tasks have been structured in the following workpackages:

WP 1: Project Management

The objective of this workpackage is to manage the project, both from the administrative and the scientific level.

WP 2: Usability attributes affected by software architecture

The objective of this workpackage is to provide a set of usability attributes, broken down into low level factors, influenced by software architecture.

WP 3: Study and description of the architecture/usability relationship.

This workpackage will output a series of guidelines that will determine the relationship between software architectures and usability in three possible, albeit not exclusive, ways:

How each architectural style supports or is detrimental to given usability attributes, improvements to existing styles to help to achieve higher usability in the software systems that use these styles, outline of a general architectural style that can be used to support usability.

WP 4: Proposal of an architecture for usability in e-commerce.

This task will select the theoretical results of WP 3 and instantiate them for e-commerce applications. This workpackage will result in an architectural pattern for usability in e-commerce systems, derived from the architectural patterns used by the industrial partners in their e-commerce applications, and their respective adaptation to include usability characteristics.

WP 5: Proposal of a software development process with usability

The result of this workpackage will be the description of a software development process specifying which usability techniques should be used for each phase.

WP 6: Development of applications for e-commerce domain

During this workpackage, each industrial partner will apply the results of WP 4 and WP 5 in a project that will serve as a pilot study.

WP 7: Dissemination and Documentation

The research will be disseminated in journals, magazines, conferences and workshops, which will include event stages for specific interest groups. Also a web page will be created containing the results of the project. The Dissemination Plan produced during this workpackage will lead the activities to do.

WP 8: Exploitation Plan

This workpackage is aimed at determining the field in which the project results will be applied. This task will not be too complicated, as one of the business areas of European and world software developers is aimed at the development of e-commerce applications, which means that the market of possible beneficiaries of the results of the project is quite extensive. It is planned for the results to be used in two ways: to improve software development by the industrial partners and as material for the consulting business area of the industrial partners. The Use Plan and the Technological Exploitation Plan will be produce from this workpackage.

WP No	Workpackage title	Lead contractor No	Start month	End month	Deliverable No
1	Project management	IHG	0	36	D1
2	Usability attributes affected by software architecture	ICSTM	0	4	D2
3	Study of the usability/software architecture relationship	RuG	5	16	D3
4	Proposal of architecture for usability in e-commerce	IHG	14	23	D4
5	Integrated development process with usability techniques	UPM	4	12	D5
6	Development of applications in the e-commerce domain	LOGICDIS	6	34	D6
7	Dissemination & Documentation	UPM	4	36	D7
8	Exploitation planning	IHG	0	36	D8

3.2 Deliverables

The deliverables for the first Review, June 17th. 2002, are highlighted in grey:

Del. No	Deliverable name	WP no.	Lead participant	Del. type*	Security **	Delivery (proj. month
D.2.	Usability attributes affected by software architecture	2	ICTSM	Report	Rest.	4
D.7.1.	Project web site (updated throughout project)	7	UPM	Web site	Pub.	4
D.7.2.1-D7.2.6	Dissemination plan	7	UPM	Report	Ist.	6, 12, 18, 24, 30, 36
D.1.8.	Project Presentation	1	UPM	Report	Pub.	5
D.5.1.	Selection of the software process and the usability techniques for consideration	5	UPM	Report	Rest.	5
D1.1.-D1.6	Periodic Progress Report	1	UPM	Report	Rest.	6, 12, 18, 24, 30, 36
D.8.1.1-D.8.1.5	Use plan	8	IHG	Report	Ist.	*6, 12, 18, 24, 30*
D. 3.1.	Effect on usability of architectural styles	3	RuG	Report	Rest.	13
D.3.2.	Improvements on architectural styles oriented to usability	3	ICTSM	Report	Rest.	13

D.3.3.	Improvements on architectural characteristics oriented to usability	3	UPM	Report	Rest.	13
D.3.4.	Report on relationship between usability and software architectures	3	RuG	Report	Rest.	16
D.4.1.	Architectural patterns for e-commerce applications	4	IHG	Report	Rest.	15
D.4.2.	Improved architectural patterns for e-commerce applications supporting usability	4	RuG	Report	Rest.	23
D.5.1.	Selection of the software process and the usability techniques for consideration	5	UPM	Report	Rest.	5
D.5.2.	Specification of the software process with integrated usability techniques	5	UPM	Report	Pub.	12
D.6.1.	Development document of pilot project by IHG	6	IHG	Report/ Prototype	Rest.	31
D.6.2.	Development document of pilot project by LogicDIS	6	LogicDIS	Report/ Prototype	Rest.	31
D.6.3.	Analysis and comparison of usability tests with and without STATUS results	6	UPM	Report	Pub.	33
D.6.4.	Final results on the relationship between usability and software architectures *(month 34)*	6	RuG	Report	Pub.	34
D.6.5.	Final results on the architectural e-commerce pattern for usability	6	ICTSM	Report	Pub.	34
D.6.6.	Final results on the integrated software process	6	UPM	Report	Pub.	34
D.1.7.	Final project report on the achievements and findings of the project	1	UPM	Report	Pub.	36
D.8.2.	Technological exploitation plan	8	IHG	Report	Ist.	36

** A short, self-evident description e.g. report, demonstration, conference, specification, prototype…*

***Int.*	*Internal circulation within project (and Commission Project Officer if requested)*
Rest.	*Restricted circulation list (specify in footnote) and Commission PO only*
IST	*Circulation within IST Programme participants*
FP5	*Circulation within Framework Programme participants*
Pub.	*Public document*

4. CONTRIBUTION TO PROGRAM / KEY ACTION OBJECTIVES

This project is related to the definition of a software architectural style (or the variation of existing styles) that supports the usability quality attribute.

In this sense, this research proposal is related to **Key Action IV**. Essential Technologies and infrastructures. As indicated in the key action objectives, “the aim of this work is to promote excellence in the technologies that are crucial to the Information Society”.

“Technologies and engineering for software, systems and services” are, as indicated in part IV.3 of this key action, crucial for the Information Society. Software applications that are crucial to the Information Society are developed by means of different development techniques. So, it is really important to provide good and tested software development techniques that can be used to produce high quality products. This project is related to software architecture activities, a very important part of the software development process, as many software product quality factors can depend on the characteristics of the architecture that supports the product.

This proposal is, in particular, related to **Action Line IV.3.1.** Software Architecture, whose objectives are “to ensure at the architectural level that required properties for the final software system will be met”. The property studied in this proposal will be the usability quality attribute. This aim is absolutely compatible with the focus of the action line “the focus is on models and notations for describing systems architectures and being able to reason about them. The main concern is to guarantee required quality attributes (for instance, scalability, performance and reliability) of systems.” In this case, the focus is on usability, as this is a crucial factor for the success of software systems, whose benefits are related not only to the user of the systems but also to the company. As described in detail later, this attribute is coming to be a strategic factor in software development businesses and, therefore, cannot be ignored.

As indicated in the 2001 Workprogramme, one of the related aspects included in the Action Line IV.3.1. is **“(iv) Domain specific architectural styles”**. Although part of the results of this project (for example, the architectural recommendations for improving software systems usability and the methodological guidelines for the integration of usability techniques into the software process) are “common to several applications”, we also provide a domain-specific architectural style to support usability, where the domain is e-commerce applications, according to the indications in the Objectives of Key Action IV.

5. INNOVATION

Architecture, and particularly software architecture, has recently become an area of intense research in the software engineering community. Over the last decade, considerable progress has been made in developing the technological and methodological basis for “engineering” architectural design. Its relevance is due to the fact that the software architecture represents the first mapping from requirements to computational components.

Hence, it contains the earliest design decisions that have the most far-reaching consequences [Clements, 96]. In this context, a software architecture can be defined as the structure or structures of the system, which comprise software components, the externally visible properties of these components and the relationships among them [Bass, 99]. In this sense, the software architecture will prescribe the structure and functionality of the future system and, therefore, *any system characteristic, and particularly quality attributes, is conditioned by architectural decisions*.

In 1996, Clements analysed the research trends in software architecture, identifying five main themes that are still in use today [Clements, 96]: architecture design and selection, architecture representation, architecture analysis, architecture-based development and evolution, and architecture recovery. Architecture design or selection focuses on the creation or selection of an architecture based on a set of functional and quality requirements. The techniques used for this aim include technology-intensive approaches describing specific architectures or another, more general method based on the idea that it is possible to document a limited set of software design templates (generally known as architectural styles), which will be used to create specific architectures [Garlan, 00]. Architecture representation focuses on the development of Architecture Description Languages in order to formally specify the nature of the components, their properties, the semantics of the connections and the behavior of the system. Examples of these languages include Adage [Coglianese, 93] and Aesop [Garlan, 94], among others. Architecture analysis focuses on the development of methods in order to evaluate software architectures regarding specific attributes. The most common attributes evaluated are related to quality attributes [Kazman, 00] [Clements, 95] [Bengtsson, 00] and to domain-specific analyses for architectures built in specific styles [Garlan, 94] [Coglianese, 93]. Research efforts considered in the area of architecture based development and evolution are led by the problem of building and maintaining a system given a representation of what is confidently believed to be a sound architecture that will solve the problem at hand [Shaw, 96] [Garlan, 00]. Other efforts focus on the description of architecture-centric system development, such as the method proposed by the SEI (Software Engineering Institute - Pittsburgh, USA) [Bass, 99]. Finally, architecture recovery focuses on the recovery of software architectures from existing systems, for instance, to evolve a legacy system [Carrière, 99]. Within this wide range of research efforts, this project focuses on the architecture analysis area. Particularly, **the project focuses on the analysis of attributes in architectural styles**, according to EU Action Line IV.3.1.

Research into the architecture analysis of quality attributes aims to bring forward critical decisions that affect different quality attributes to the architectural design phase. For this purpose, this line of work is based on the development of methods of analysis and evaluation of architectures that can be used to evaluate architectural styles or specific architectures in respect of these attributes. Briefly, this line of work is founded on understanding relationships between achieving desired software quality attributes and software architectures [Bass, 99]. For example, one of the quality attributes under study is performance. Originally linked to coding and low-level design, efforts are now being made to support this characteristic at the time of architectural design. For this purpose, different architectural styles are being analyzed in relation to this quality attribute. There is a desire to determine relationships between other quality attributes and architecture, as has been done with performance. Work on analyzing quality attributes in software architectures is been carried out by several researchers. One of the precursors of these studies is the SEI. This Institute is now working on the development of a software analysis method (Architecture Tradeoff Analysis Method$_{SM}$, ATAM$_{SM}$) [Kazman, 00], as well as on the elaboration of Attribute-Based Architectural Styles (ABASs) [Klein, 99]. Each ABAS provides guides for developing and analyzing the software architectures of a system, considering one quality attribute at a time. The quality attributes now being studied in relation to software architecture are: performance, modifiability, reliability and availability [Klein, 99] [Bass, 99]. On the other hand, *usability is one of the quality attributes considered critical nowadays*. Usability is actually a quality attribute already considered in different classifications of quality attributes [IEEE, 98], [ISO, 91], [Boehm, 78]. Software systems usability is important for several reasons [Trenner, 98] [Donahue, 01]: it improves employee productivity and raises team morale, reduces training and documentation costs, increases ecommerce potential, etc. The cost savings of applying usability techniques have also been made apparent in numerous cases [Nielsen, 93] [Cooper, 99]. In sum, it can be said that usability has its place among the quality strategic factors in software development.

As mentioned earlier, a software architecture is the earliest life-cycle artifact that embodies significant design decisions. This means that an architecture can either allow or preclude the achievement of most of a system's targeted quality attributes, including usability. It is widely recognized [Nielsen, 93] [Constantine, 99] that design decisions drastically influence the final usability level of a system. Indeed, there is quite advanced ongoing research that focuses on defining patterns that promote usability [Mahemoff, 98]. However, this research centers on low-level design and falls outside the area of software architecture research. Note that despite the importance of usability for software systems, its relationship with software architectures has only started to be studied recently. Indeed, the only research we know that has this goal was launched by the architectures group (Architecture Tradeoff Analysis Group) at the SEI in 2000 [John, 00]. This research has reached the point of recognizing an explicit relationship between software architectures and usability. Indeed, Bass [Bass, 01] claims *"Usability is architectural and can be addressed via a proactive strategy"*. At present, the SEI architectures group is working on the identification of software architecture-sensitive aspects of usability to be considered in the analysis of specific software architectures in the future [Bass, 01]. So far, they have achieved a preliminary identification of architecture-sensitive usability attributes, which are:

- Individual Human Effectiveness
- Efficient Routine Performance
- Error-free Performance
- Tolerating
- Human Errors
- Non-routine Performance
- Problem-solving
- Learning
- Routine Performance
- Preventing
- Tolerating
- User satisfaction
- Systems Errors
- Preventing
- Tolerating

Although the fact that the SEI has highlighted the existence of an explicit relationship between usability and software architectures could be basis enough for considering the research proposed in this project as feasible, a simplified example is presented below showing how considering usability at the time of architectural design can improve the quality of the final system in respect of this attribute. This example will focus on enhancing usability related to user satisfaction, specifically the user feedback attribute, which contemplates providing the user with an explicit model of how the system is working.

Suppose that a software system uses an instance of the pipeline architectural style. This style is characterized as shown in Figure 1, where the user enters a system input, the system processes this input and returns the respective output to the user. Imagine that the process to be performed by the system is complex and takes several minutes. In this case, the feedback attribute is important for improving user satisfaction.

One way of considering this attribute at the time of architectural design is to add a module designed to inform the user about the system progress (Figure 2). This module can either provide simple messages to the user indicating the percentage of work completed or provide more sophisticated models where the user can query the type of tasks that are being performed, their utility, relationship with other tasks, etc.Note that the fact that this decision is included in the architecture of the system raises the usability of the system. Not considering this decision at this time does not mean that the system will not address this attribute, but that the decision is left until later on in the development process, for example, low-level design or programming. The satisfaction of this attribute will then depend on how concerned the professionals responsible for these tasks are about usability. Thus, the consideration of the characteristics of an architecture that supports this attribute at the time of architectural design assures that these characteristics will necessarily have to be addressed at later phases in the development process and, therefore, that the final product will be more usable.

This is the aim of this project –to study how specific software architectures can benefit the different aspects of usability to determine which of these attributes are affected by architectural decisions and, especially, how, so that improvements can be made at the architectural level to improve usability. Therefore, **the innovation of this proposal** in relation to ongoing architecture research can be said to be **the study of the relationship between the usability quality attribute and software architecture**, as there is currently only another group (at SEI) starting to research this subject.

Although the SEI has already started to study the relationship between usability and software architectures, **this project proposes innovations in the approach to the solution**. The SEI has a long tradition as consultants for software development companies. In this context, the SEI architectures group bases its activity on providing advice about certain quality attributes in specific architectures provided by the companies. For this purpose, they use what are known as scenarios, which represent specific uses of the software systems that contain the architectures under evaluation, and analyze whether or not these architectures support the quality attributes under study. The approach presented in this project goes further and proposes, as mentioned before, **to identify a series of characteristics for addition to existing architectural styles to support usability** and even, if feasible, to define a new architectural style that supports usability. That is, we aim to output a set of architectural patterns that support different usability aspects. With regard to the objective of integrating usability techniques into the development process, it is also an innovative proposal. Although usability is accepted as a desirable quality attribute, the aim of developing usable software has been impeded by the difficulty of employing usability techniques along with software engineering techniques, as usability techniques consider topics related to fields like cognitive psychology, ergonomics or human factors, and sociology [Windl, 01].

The integration of usability in routine development (this special issue was promoted by the UPM responsible researcher who was one of the guest editors). It is surprising that all the papers that addressed the subject came from companies that are making the effort at integration without research results to smooth the way. Therefore, there is a deficit in this field, despite the need on the part of the industry (and especially the software and Internet industry). The only attempt to integrate some usability techniques into the development process was addressed in [BIUSEM, 95]. However, this integration affects only the requirements task and interface design, omitting the influence of usability on the other phases of the development process. **It is innovative to propose an entire development process that would integrate traditional software development techniques along with usability activities to completely bridge the two disciplines**.

As mentioned above, usability is particularly important in e-commerce applications. This is why the results of this project will be particularly focused on this kind of projects in which the industrial partners, IHG and LogicDIS have lengthy experience.

6. REFERENCES

*[Bass, 99] Bass, L., et al. “*Architecture-Based Development*”. CMU/SEI-1999-TR-007. SEI/CMU, 1999.*

*[Bass, 01] Bass L and John B. “*Achieving Usability through Software Architecture*”. Key talk at the Human Computer Interaction, Consortium, February 2001.*

*[Bengtsson, 00] Bengtsson, P., et al. “*Analyzing Software Architectures for Modifiability*”. University of Karlskrona. Technical Report HKR-RES-00/11-SE. 2000.*

*[BIUSEM, 95] BIUSEM Consortium. “*Benefits of Integrating Usability and Software Engineering Methods*”. Final Report ESSI Project 10290, 1995.*

[Boehm, 78] Boehm B., et al. “Characteristics of Software Quality”. North Holland, 1978.

*[Carrière, 99] Carrière, S.J., et al. “*Software Architectural Transformation*”. Proceedings of the Sixth Working Conference on Reverse Engineering. Atlanta, Georgia. October 6-8, 1999.*

*[Clements, 95] Clements, et al. “*Predicting software quality by architectural evaluation*”. Proceedings of the Fifth International Conference on Software Quality. Austin, USA. October 23-26, 1995.*

*[Coglianese, 93] Coglianese, L., et al. “*DSSA-ADAGE: An environment for architecture-based avionics development*”. Proceedings of AGARD’93, May 1993.*

*[Constantine, 99] Constantine L. L., et al. “*Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*”,Addison Wesley Longman, Reading, Mass., 1999.*

[Cooper, 99] Cooper, A. “The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity”,Sams, Indianapolis, 1999.

*[Donahue, 01]Donahue G. M. "*Usability and the Bottom Line*". IEEE Software, vol. 16, n. 1. pp 31-37.*

*[Garlan, 00] Garlan, D. “*Software Architecture: a Roadmap*”. Proceedings of the 22nd International Conference on The Future of SoftwareEngineering (ICSE’00). Limerick, Ireland. June 4-11, 2000.*

*[Garlan, 94] Garlan, D., et al. “*Exploiting style in architectural design environments*”. Proc.of the Second ACM SIGSOFT Symposium on Foundations of Software Engineering (SIGSOFT’94.). December, 1994.*

*[IEEE, 98] IEEE, “*IEEE Std 1061: Standard for a Software Quality Metrics Methodology*”, 1998.*

*[ISO, 91] ISO/IEC, “*ISO 9126: Information Technology - Software quality characteristics and metrics*”, 1991.*

*[John, 00] John B. "*Working Notes: Usability Benefit Hierarchy*". Working Report CMU/SEI. June 2000.*

*[Kazman, 00] Kazman, R., et al. “*ATAM$_{\text{SM}}$: Method for Architecture Evaluation*”. CMU/SEI-2000-TR-004. Software Engineering Institute, Carnegie Mellon University, 2000.*

*[Klein, 99] Klein, M., et al. “*Attribute-Based Architectural Styles*”. CMU/SEI-99-TR-022. SEI/CMU, 1999.*

*[Nielsen, 93] Nielsen J. “*Usability Engineering”*, AP Professional, Boston, Mass., 1993*

*[Mahemoff, 98] Mahemoff, M.J., et al. “*Pattern Languages for Usability: An Investigation of Alternative Approaches*”. Proc. of the Asia-Pacific Conference on Human Computer Interaction (APCHI’98), Japan, July 1998.*

*[Shaw, 96] Shaw, M., et al. “*Software Architecture: Perspectives on an Emerging Discipline*”. P. Hall, 1996.*

[Trenner, 98] Trenner L., et al. The Politics of Usability. *Springer, London, UK, 1998.*

*[Windl, 01] Windl H., et al. "*Usability Basics for Software Developers*". IEEE Software, vol 16, n. 1, pp 22-29.*