Evaluating Web Usability With MiLE+

Luca Triacca
TEC-Lab
University of Lugano, Switzerland
luca.triacca@lu.unisi.ch

Alessandro Inversini
TEC-Lab
University of Lugano, Switzerland
alessandro.inversini@lu.unisi.ch

Davide Bolchini
TEC-Lab
University of Lugano, Switzerland
davide.bolchini@lu.unisi.ch

Abstract

Websites (or more in general interactive applications) need to be evaluated from a usability perspective for improving their quality for the end-users. This paper introduces a systematic methodology called MiLE+ which proposes an innovative approach to usability evaluation under several aspects. The main original feature is the separation between application-independent analysis (based on usability principles) and application-dependent analysis (based on the requirements of the application). The paper explores the differences between these types of analysis and presents the activities for carrying them out in an effective way.

1. Introduction

Complex web applications, whose goal is to communicate information and services to a large number of users, have to pay special attention to their usability, or rather quality. Clearly, this is an arduous task for the designers (and in general for all the stakeholders involved in the development of the application): web applications are of growing complexity, address several targets, deal with complex content and have different communication goals: for all this reasons, they need to be well "usable" and efficient.

Establishing the quality means to take into account the degree of satisfaction that the users have during the interaction with the web site. The most important "unit of measurement" of satisfaction is usability, intended as "the effectiveness, efficiency, and satisfaction with which specified users achieve specified goals in particular environments" (ISO 9241 definition).

Starting from proven practices and approaches in Human-Computer Interaction (HCI) and usability engineering, this paper presents an innovative methodology, called MiLE+, for the usability evaluation of hypermedia and interactive applications.

2. Related Works

There are different approaches that can be applied in evaluating usability of a web application; basically all these techniques can be divided into two main categories: usability inspection methods and empirical Testing. *Usability Inspections methods* (also called "expert review" methods) is the generic name for a set of methods based on having expert evaluators instead of final users inspect or examine usability-related aspects of a user interface [1]. During the inspection phase, the expert evaluator goes through the application looking for usability breakdowns; the expert judges the application thanks to her/his personal skills and competences, but s/he also has a set of instruments (such us usability principles, criteria or a set of previously defined guidelines) that are applied during the inspection. The main advantage of usability inspections methods is the relationships between costs and benefits. In fact, to perform usability inspection does not require any special equipment and the inspector alone can detect a wide range of usability problem in a little time. Some inspection techniques are: *Heuristic evaluation* [1] and *Pluralistic Walkthrough*. During the heuristics Evaluation, usability specialists have to judge whether each dialogue element conforms to established usability principles or not. Instead, during a *Pluralistic Walkthrough* a group of inspectors (composed of different stakeholders involved in the design) "walks" through the web site, performing the major plausible tasks, trying to touch every page likely to be used [2]. As drawbacks, these two methods generally focus on "surface-oriented" features of the graphical interface (mainly at page level) [3]. Only few of them address the usability of the application structure, i.e., the organization of both information elements and functionality. Moreover, they are strictly dependent on the individual know-how, skill and judgment of inspectors [4].

Empirical Testing methods, (also called User based methods) investigate usability through the observation of the user interacting with the application. In fact usability properties are assessed by observing how the system is actually used by some representatives of real users [5], [6]. Some techniques in this category are *thinking aloud* and *contextual inquiry*. During a *thinking aloud* the user should think aloud while performing some specific task with the system. By verbalizing his thoughts, the user allows the observers to know his opinions and feeling about the application. Instead, Contextual inquiry is a specific type of interview for gaining data from the user. The aim of this technique is the understanding of the context in which the application is used.

As it is possible to image, some problems arise using empirical methods. The main problems are related to the difficulty to properly select representative user samples, and to the difficulty to adequately train them to manage also advanced functions of a website [4]. Finally, it is difficult to reproduce actual situations of usage in a limited amount of time. The difficulty of reproducing real conditions is called "Hawthorne effect" [7]: if the variables of the experiment are manipulated, it is possible that the productivity of the group observed decreases.

Within these two categories (User Testing and Inspection Methods) the most current usability evaluation techniques for web applications are alternatively based on two main approaches: Heuristic-driven evaluation and task-driven (scenario-driven) evaluation.

In the *heuristic-driven evaluation* checklists and usability principles are used [8]. The main drawbacks related to this methodology refer both to the usability principles inspiring the reviewer which are very good for detecting problems but provide poor design suggestions for the re-design; on the other hand, heuristic is very effective for measuring usability qualities of the site but captures very hardly the evaluation of complex scenarios.

Task-driven evaluation provides sets of tasks guiding the user testing, walkthrough and inspection techniques [9], [2]. Normally, the evaluation based on tasks is used within a scenario, that is, the description of a concrete episode of use of the application, a "story about use" [10], [11]. This methodology has some disadvantages, in particular Scenario-based approaches can easily detect the feasibility of a task, i.e. whether a task can be actually accomplished or not but thy do not identify what exactly caused the failure or the success of the task. All the methodologies presented above have been intended to work alone and alternatively. However, it seems that for performing an accurate usability inspection, these approaches must be mixed together to mitigate their respective drawbacks and to exploit the advantages of each single methodology.

3. MiLE+ method

MiLE+ is the fruit of common research performed by TEC-Lab (University of Lugano) and HOC-Lab (Politecnico of Milan). MiLE+ tries to mix together some features of the methods above presented stressing the strengths and minimizing the drawbacks. In particular, MiLE+ is a usability inspection framework for web applications that strikes a healthy balance between heuristic evaluation and task-driven techniques. Clearly MiLE+ is not merely the sum of the more interesting characteristics of other methods, but it also introduces a new conceptual approach and several tools.

3.1 Separating Application-Independent and Application-Dependent Analysis

The first conceptual innovative feature is to distinguish between the application-independent and the application-dependent analysis. Indeed, an interactive application can be evaluated from one hand from a technical and more "objective" perspective, and on the other hand the evaluation can be situated in the context of use of the application, to assess how it meets user goals.

3.1.1 Application Independent Analysis. Every human artifact can be observed, analyzed and evaluated from an objective and generic point of view. For example, if we think on a chair, this should have some technical characteristics for making it usable (for example a comfortable back of chair, a stable bearing…). If we consider the chairs below, the first one (Figure 1) is a usable one from a technical point of view: it has a comfortable back, a stable bearing, and two relaxing arms: from an objective analysis it is not possible to state that it is not usable. On the contrary, the second chair (Figure 2) is not a usable chair: the support is not stable at all (just one leg), there is no a chair back, no arms.

Figure 1: a usable chair

Figure 2: a non-usable chair

An interactive and multimedia application, in particular websites, can be analyzed from an objective point of view as well as a chair. Clearly they are products having different levels of complexity, but the conceptual approach to the usability evaluation could be similar. Indeed, there are technical usability aspects that can be evaluated independently from the application under analysis (the term technical is used in a broad sense, not only referred to the technology behind the application). Making an Application Independent Analysis means to analyse the features that can be evaluated even without knowing the purposes and the users of the application. There are technical aspects that should comply with general usability parameters (heuristics). In this sense, these types of features are related to design aspects that can be considered without involving the users in the design. In fact, there are several usable design strategies which could be used without thinking to particular users.

Let us see some examples of application-independent features in websites and related usability problems:

Background contrast: independently from the type of website we are using, the contrast between the background and the text should allow the legibility of the textual content.

Figure 3: lack of contrast between background colour and font

This example, excerpts from MOCA website (www.moca.org) presents a lack of contrast between background and text. The low legibility of the text is a problem independent from the application we are using.

"Go back" (Backward Navigation) in the navigation starting from an index/list: when the user reaches a list s/he has to control the navigation while going from the starting index to each element and while going back from one element to the index.

Figure 4: once the user reaches the painting s/he can not go back to the list

In the case of the Guggenheim Museum website (www.guggenheimcollection.org), once the user reaches the list of art's works now on view and select a painting (e.g. Georges Braque - Landscape near Antwerp) he achieves the selected page correctly. When the user tries to return to the list of art's works the backward mechanism is absent. The only navigational mechanism are two links called "Previous Braque work" (1) and "Next Braque Work" (2) that allow navigating within a guided-tour of the Braque's work. Evaluating this navigational feature we do not consider the back of the browser, see that is a stand-alone application and it is not part of the website's

design (besides sometimes the back of the browser has anomalous behaviour).

3.1.2 How to evaluate application-independent aspects? The activity for performing the application independent analysis provided by MiLE+ is called Technical Inspection. The aim of MiLE+'s Technical Inspection is the identification of design problems and implementation breakdowns. The output of this evaluation is a number of "technical" problems that are application independent (e.g. the fact that the font size of a text is too small – graphic technical problem – it is a problem independent from the type of application). During this analysis the evaluator examines the web application taking into account a number of design dimensions, assuming the point of view of the designer and not of the end-user. The design dimensions are:

- *Navigation*: the website's structure
- *Content*: information provided by the application,
- *Technology/Performance*: technological performance of the application.
- *Interface Design*: this is a broad dimension that includes semiotics, graphics (graphical design and layout) and cognitive (what the user learns about the application and its content)

During the Technical Inspection problems are discovered using the heuristics checklists (selected from the library of technical heuristics) and scenarios: these two elements compose the Usability Kit (U-KIT) for Technical Inspection. It is important to underline that the use of scenarios are not mandatory. Indeed, we do not evaluate the adequacy of scenarios, but they are useful for navigating with clear goals within the application (so the inspector can concentrate his evaluation on the most important parts of the website). However, the most important tool for Technical Inspection is the technical heuristics' library divided by design dimension. In the Table 1 above some examples of technical heuristics are presented. Actually the technical heuristics' library is composed of 36 navigational heuristics, 8 content heuristics, 7 technology/performance heuristics and 31 interface design' heuristics (a total of 82 technical heuristics.)

Table 1: some examples of Technical Heuristics

Dimension		Examples of Heuristics
Navigation		Consistency of the overall navigation
		Control of a guided-tour
Content		Text accuracy
		Multimedia consistency
Technology/Performance		System reaction to errors of a user
		Operations management
Interface design		
	Cognitive	Information overload
		Scannability
	Graphics	Font size
		Text layout
	Semiotics	Ambiguity of string of characters
		Conventionality of interaction images

The use of dimension as aspect under analysis and Technical Heuristics as "unit of measure" is partially comparable to GQM model [13], where Goal are the dimensions under analysis (e.g. Content, Navigation…) and Questions are Heuristics to be evaluated (e.g. How the structural navigation works?). See that MiLE+ is a qualitative analysis metrics are not included.

3.2 Application Dependent Analysis

As described before, it is possible to analyze the application taking into account the context of use of the application. During the Application Independent Analysis the inspector evaluate the application out of its context. On the contrary when he performs the Application Dependent Analysis he has to situate the evaluation within different scenarios of use (or situation of usage).

If we think on the chairs' example made before, it is possible to evaluate them taking into account the scenario of use. Shortly, the scenario of use of the first chair (the office's chair) is a situation where people need a comfortable chair (they have to stay sitting for more then 8 hours), a chair that can easily displace the people within the office… Considering this scenario of use the first chair remains usable. The second chair (a milking stool), which is not usable from a technical point of view, is used in a very particular scenario: a farmer which have to milk several cows. Situating the chair within this scenario it is possible to state that it is a usable chair as well. Indeed, the milking stool allows the farmer to achieve his objectives. However the chair remains lacking in technical usability and it could be improved.

Figure 5: a usable office's chair

Figure 6: a usable milking stool

An interactive application (more than a chair!) which addresses several users should be also evaluated taking into account the scenarios of use (the concept of scenario will be described in depth in the next paragraphs). During the Application Dependent Analysis the evaluator has to determine if the user(s) are put in the right conditions for achieving its (their) goals. Verifying the capability of the user to reach his/her goals means to answer questions such as: Do people find the information they need? Are people properly driven and guided to unexpected content? Is the content relevant for the user(s)? Is content enjoyable/entertaining for the users?

Besides, it is also very important to evaluate if the application can be effectively used in a specific context (while driving, while at home, office, walking, visiting…). Understanding users, their goals and the contexts of use is essential to evaluate the application dependent usability.

For explaining in depth the features related to the application dependent analysis and relative usability problems, we present some examples:

Multilinguisticity: the content addressing to different type of users speaking difference languages should be given in more than one language. The multilinguisticity is a feature strictly related to the scenarios of use of the application and to its requirements. It is not possible to state that multilinguisticity is a technical usability feature, because the choice of implementing more then one language in a website is strictly dependent on its target audience.

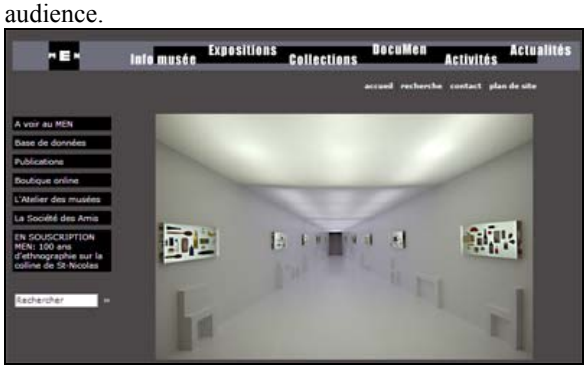

Figure 7: home page of MEN website (www.men.ch)

The "Musée d'ethnographie de Neuchâtel" (MEN) website (www.men.ch, see Figure 7), most of information are provided only in French, even though it is presumable that the audience is not only local, but also international (one of the possible target are cultural tourists). The lack of multilinguisticity creates a usability problem related to the contents' fruition for a specific target (cultural tourist).

Predictability: is the capability of interactive elements (symbols, icons, textual links, buttons, images…) to anticipate the related content and the effects of the interaction. The semantics and the semiotics of the interactive elements (e.g. links labels) are strictly related to the type of users that will use the application. For example, if we develop a CD-Rom about Michelangelo addressed to Children, the link labels should be understandable for the Childs (they should be able to anticipate the related content, which page they will reach).

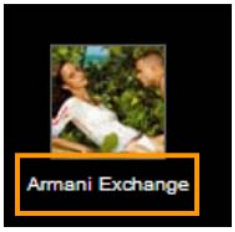

Figure 8: this label is not clear.
Which is the content behind it?

Using the Armani website (www.armani.com, see Figure 8) one of the link labels is called "Armani exchange". It is not clear at all which is the content behind this label. Only a user which knows in-depth Armani, knows that Armani Exchange is one of the Armani's Collection. From a usability point of view this becomes a problem if, for example, the intended users of the website are not only "Armani's fan", but also people which are just curious (they do not have the background for understanding this label). Also this type of feature and related usability problem are strictly dependent from the type of application.

3.2.1 How to evaluate application-dependent aspects? The User Experience Inspection and the Scenario-based User Testing are the activities for performing the application-dependent evaluation. The User Experience Inspection is a scenario-based inspection which allows understanding the existence of application-dependent problems. This means that the evaluator has to imagine stories of use. For this reason, he has to set-up the "User Experience" KIT tailor-made for the application under analysis. The KIT is composed by:

Scenario library: for creating a domain's library the inspector has to interact with different stakeholders: the client, domain experts, end-users, etc.

Table 2: example of scenario used for the evaluation of a museum website

Scenario description	Well-educated American tourist who knows he will be in town, he wants visit the real museum on December 6th 2004 and therefore he/she would like to know what special exhibitions or activities of any kind (lectures, guided tours, concerts) will take place in that day.
User profile	Tourist
Goal	Visit the Museum in a specific day
Task(s)	• Find the exhibitions occurring on December 6th 2004 in the real museum • Find information about the museum's location

Library of User Experience Indicators: during the User Experience Inspection the evaluator has to put himself in the "shoes of the (different) users". This means that he has to examine the relevant scenarios using some criteria called User Experience Indicators. These criteria are divided in three categories corresponding to the different types of user interaction experiences. These categories are:

- *Content Experience Indicators*: measure the quality of user interaction with the content of the application.
- *Navigation & Cognitive Experience Indicators*: allow the measure of how the navigation works and the cognitive aspects of the application meet the cognitive world of the user(s).
- *Interaction Flow Experience Indicators:* permit the measurement of how the interaction with the application is appreciated by the users.

Table 3: Examples of User Experience Indicators

Categories of interaction	Examples of User Experience Indicators
Content Experience	Completeness
	Relevance
	Comprehensibility
Navigation & Cognitive Experience	Predictability of interactive elements
	Learnability
	Memorability
Interaction Flow Experience	Naturalness
	Engagement
	Recall

Actually the User Experience Indicators library is composed of 7 Content Experience Indicators, 7 Navigation&Cognitive Experience Indicators and 6 Interaction Flow Experience Indicators, that means a total of 20 Indicators.

It is also possible to compare the User Experience Inspection to GQM model [12], where Scenarios are the Goal (the object of the evaluation) and the User Experience Indicators the Questions to answer (e.g. How is it complete this text? How relevant?).

The User Experience Inspection is strictly related to the **Scenario-based User Testing.** Indeed, the main goal of the Scenario-based User Testing is to empirically validate or invalidate the results provided by the User Experience Inspection. During the test the user accomplishes several tasks belonging to the critical scenarios identified in the User Experience Inspection. Indeed, performing the User Experience Inspection before the Scenario-based User Testing allows selecting the significant scenarios of use to check with the user testing. Their importance could be related to two main factors:

Number and gravity of problems identified performing a specific scenarios: if a scenario presents several problems with a high gravity, testing it with end-users allows a double check on the usability issues.

Relative importance of a scenario for the application: if a scenario is very important for the application, it should be useful to test it with end-users for having a double control on the quality of the scenario.

3.3 The MiLE+ Evaluation Process

In this section we will briefly present the MiLE+ evaluation process. It is important to underline that the preliminary activities made by the inspector are the choice of the evaluation activity to perform (Technical Inspection, User Experience Inspection or both) and the creation of the U-Kit which he will use.

3.3.1 Application-dependent analysis: evaluating technical features. The Technical inspection has two steps:

1. *Performing the (selected) tasks within a scenario (not mandatory) or random inspection*
2. *Evaluating the application using Technical Heuristics*

1. Performing the (selected) tasks (not mandatory) or random inspection: the first activity performed by the inspector is to decide if he will perform an inspection using scenarios (which allows concentrating the analysis on the most important areas of the website) or carry out a random inspection (without taking into account particular task and scenarios).

2. Evaluating the application using Technical Heuristics: once started the analysis the inspector has to complete an evaluation matrix, giving both a score for the selected heuristics (he has to decide the scale) and a comment for each score.

Table 4: Example of technical matrix (scale: 3 poor, 6 sufficient, 9: good)

Dimension	Heuristic	Score	Comment
Content	Conciseness	3	The text is to long and it is not easy to read.
	Text errors	9	The text does not present errors.
Navigation	Orientation in three navigation	6	Sometimes it happens that if we pass from one section to another, we do not find orientation clues.

3.3.2 Application-dependent analysis: evaluating the Scenarios. The Evaluation of each scenario has three steps:

1. *Performing the (selected) tasks*
2. *Evaluating the tasks through User Experience Indicators (UEIs)*
3. *Weighting the results according to user profiles and communication goals*

1. Performing the (selected) tasks: the goal of this activity is to assess the feasibility of some "critical" tasks. According to salient user scenarios, the inspector defines a set of tasks and performs them on the site. For each task, the reviewer assesses whether or not it can be properly accomplished.

2. Evaluating the tasks through User Experience Indicators (UEIs): the inspector has to check a list of UEIs concerning the different facets of usability/quality (e.g. richness, completeness, etc.). For each indicator (in relation to a specific task), a score must be given. The output of this activity is a task matrix which reports the scoring (of each attribute) and the result obtained by every task.

3. Weighting the results according to user profiles and communication goals: in this phase the inspector has to establishing the "real quality" of each critical task with respect to their relevance. After the scoring phase is over, the set of collected scores is analyzed through "weights" which define the relevance of each indicator for a specific user scenario. Weighting allows a clean separation between the "scoring phase" (use the application, perform the tasks, and examine them) from the "evaluation phase" in a strict sense, in which the applications' and the stakeholders' goals are considered. The result is a final matrix that shows the overall results obtained by every task. This matrix reports the results according the goal and the requirements of the application.

Table 5: Example of user experience evaluation matrix for a single task

	UEIs				
Task: Find information about the history of museum collection	Predictability	Understandability	Richness	Comprehensibility	**Global Score for this Task**
Scores	8	8	5	6	6.75 (just average score)
Weights	0.1	0.1	0.5	0.3	
Weighted Scores	0.8	0.8	2.5	1.8	5.9 ("weighted average")

Note: the Scale used for completing the analysis is:

- Scores: 0-10 (0: bad, 10 very well done),
- Weights: 0-1 (0: UEI not important; 1 very important. The sum of weights does not be more than 1)

The website under evaluation obtained a pass mark for this task (5,9/10). Analysing carefully the partial results, it is evident that both the *richness* and *comprehensibility* of the information regarding the collection's history should be improved (they are the more important UEIs for this scenarios – weight 0.5 for *richness* and 0.3 for *comprehensibility* – and they have obtained a quite negative judgement – 5 and 6).

3.4 Advantages of separating Application Independent Analysis and Application Dependent Analysis

The necessity of separating the application-independent and the application-dependent analysis is related to the different typology of the problems and consequently to the needed resources for analyzing and correcting them. Performing an application-independent usability evaluation needs less time with respect to the application-dependent evaluation and provides results which are more reliable. Indeed, most results obtained during the application-independent analysis are almost unquestionable (for example an unreadable text is always a problem independently from the application under evaluation). However, in accordance to the ISO 9241 definition the "real" usability evaluation is made performing the application-dependent analysis (both during the User Experience Inspection and Scenario-based User Testing). Indeed, during this analysis we take into account particular users, trying to accomplish their goals in an effectively, efficiently and satisfactorily way in particular environment. However, it is important to underline that the evaluation process for analyzing, discovering and solving application-dependent problems, is more complex. Indeed, the

problems' analysis and detection needs a preparatory phase for setting all the different tools (e.g. the creation of scenarios, the selection of the User Experience Indicators to use…; in the case of user testing the recruiting and the screening of the participants…). Besides, see that these types of problems are strictly connected with the application's nature, its goals, its users and its domain, the problems' correction needs a deep work which involves not only the development team but also other stakeholders (that's end-users, directors and managers). So, correcting them is more complicated then the resolution of technical problems and it is more expensive in term of invested resources: Taking again the example of MEN museum which is only in French (multilinguisticity user experience problem) the process for solving it passes through the director of the museum, the curator, the development team, the translator…: the process needs a lot of resources.
Summarizing, the main advantage of separating application-dependent and application-independent analysis is the possibility to perform the evaluation take into account two main constraints: resources at disposal (temporal and economical) and the knowledge of the application's domain. It is important to underline that sometimes it happens that the results obtained performing the application-independent and the application-dependent analysis could be marked by conflict. Indeed, it could be happen that even if the inspector discovers technical issues, the evaluation of a scenario obtained good results. In this case, the inspector should however communicate the technical issues. In the chairs' example made before, even if the milk stool is usable in a particular scenarios, it would be possible to improve some technical features (e.g. the comfort). In the case of web applications even though a scenario is well judged, it could be possible that some technical problems are discovered (e.g. the font size to small). So, the inspector has to manage possible conflicts in findings and he has to communicate separately the results of each activity and suggest the requirements for improvement considering these different aspects.

4. Conclusions and Future Work

In this paper we have presented some features of MiLE+; particularly it has been highlighted the importance of the application dependent and application independent analysis, with the different activities involved in each moment of the evaluation. We also have presented several "ready to use" tools, such as the heuristics' library and the user experience indicators.

Concerning the re-usability of the method, it is important to underline that MiLE+ is used every year by more then 100 University students (both at bachelor and master level), researchers and practitioners.

At now, TEC-Lab is developing a specific U-KIT for Cultural Applications (e.g. museum websites, digital libraries…) and the future work will focus on the development of U-KITs for others domains (e.g. banks, tourism, educational…). Another future work will be the expansion both of the heuristics and user experience indicators.

Besides, it is important to emphasize that the methodology is continuously tested thanks to the interaction with usability experts and designers in order to understand the MiLE+'s breakdowns.

5. References

[1] Nielsen, J., Mack, R., "Usability Inspection Methods", Wiley 1994

[2] Brinck, T., Gergle, D., Wood, S.D., "Usability for the web, Morgan Kaufmann", 2002.

[3] Green, T.R.G. & Benyon, D.R., "The skull beneath the skin; Entity-relationship modeling of information artefacts", int. J. Hum.-Comput. Stud., vol. 44, no. 6, pp. 801-828, June, 1996

[4] Matera, M., Costable M.F., Garzotto F., Paolini P., "SUE Inspection: An Effective Method for Systematic Usability Evaluation of Hypermedia", IEEE Transaction, Vol.32, No. 1, January 2002

[5] Whiteside J., Bennet J., and Holtzblatt K., "Usability engineering: Our experience and evolution, in Handbook of Human-Computer Interaction", Ed. Amsterdam, The Netherlands, North-Holland, 1988, pp.791-817

[6] Dix, A., Finlay, J., Abowd, G., and Beale, R., "Human Computer Interaction", 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1998

[7] Roethlisberger, F.J. and Dickson, W.J. "Management and the Worker", Boston, Mass., Harvard University Press, 1939

[8] Nielsen, J., "Designing web Usability", New Riders, 1999

[9] Rosson, M.B., Carroll, J., "Usability Engineering", Morgan Kaufmann, 2002

[10] Cato, J., "User-Centred web Design", Addison Wesley, 2001

[11] Carroll, J., "Making Use – Scenario-based design of Human-Computer Interactions", MIT Press, 2002.

[12] Basili, V.. "GQM approach has evolved to include models", IEEE Softw. 11, 1, 8, 1994.