An Applied Ergonomics Study on IT User Interaction in a Large Hydroelectric Company in the Northeast of Brazil

Marcelo Márcio Soares[1], Fabio Campos[1], Walter Correia[1], André Neves[1], Joao Corte[2], and Saul Mendonca[2]

[1] Post graduate Programme in Design, Federal University of Pernambuco, Brazil
[2] CHESF – Companhia Hidro-Elétrica do São Francisco
marcelo2@nlink.com.br, {fc2005,andremneves}@gmail.com, ergonomia@terra.com.br, {jcorte,saulsm}@chesf.gov.br

Abstract. The aim of this study was to develop an ergonomic methodology to verify the compliance and usability of software, in line with, ISO 9241, which may assist software developers, maintainers and assessors in a power distribution utility in northeastern Brazil to improve the quality of Human-Computer Interactions. It also set out to develop and implement a software tool that might incorporate the methodology developed, thus enabling IT professionals to conduct a compliance review with a view to increasing productivity and reducing failures of the services managed or implemented by using this software which had undergone ergonomic verification.

Keywords: Ergonomic methodology, ergonomic analysis software, usability testing.

1 Introduction

Currently there is awareness that there is a constant race in the field of systems that are used interactively to present users with artifacts which are extremely up-to-date and compatible with their needs. Also, it can be seen that there is a tendency to try to create demands that go beyond consumers' needs. Aside from this aspect of marketing, there are studies that aim at seeing to it that the potential of such artifacts is best taken advantage of by those who acquire them. The huge problem is that it is very common for these types of artifacts to be launched without properly testing and analyzing how they will be used which results in usability problems at some time in the future.

At bottom, Ergonomics is dedicated to analyzing systems with a focus on the users. To do so, a body of knowledge needs to be put together that reaches synthetic conclusions, based on an analysis in which the "success" or "failure" of a task is detected. Therefore, what is essential for the study of systems and software from the perspective of Ergonomics is that there should be an assessment from the point of view of information (cognitive ergonomics), due to the fact that these systems should

A. Marcus (Ed.): DUXU/HCII 2013, Part IV, LNCS 8015, pp. 113–120, 2013.
© Springer-Verlag Berlin Heidelberg 2013

include, inter alia, the principles of ease of learning, clarity of information and comprehensibility.

The aim of this study is to develop an ergonomic methodology for verifying the compliance and usability of software, in line with ISO 9241, which may assist software developers, maintainers and assessors of a power distribution utility in the Northeast of Brazil to improve the quality of Human-Computer Interactions. To do so, a software tool was developed and implemented that might incorporate the methodology which is the object of the research and enable the company's IT professionals to conduct analyses on the compliance of systems that the company uses. Thus, a computer system was developed which is based on the techniques and heuristics of the usability of computer systems to support the process of ergonomically evaluating software, such as those that target supervising, controlling and protecting electrical systems.

2 Theoretical Foundations

The popularization of the use of the computer as a work tool has been demanding that ever-greater care be taken over the quality and efficiency of the Human-Computer Interface (HCI) since the productivity, health and physical and mental well-being of workers operating these tools have a direct relationship with the interface of software and computers with which they work.

Human-Computer Interaction is the communication instrument through which the user interacts with the computer application; on which may depend, potentially or directly, the wholeness of people (clients and employees), the quality of services, and significant losses or profits. Ergonomic problems, and problems of usability in particular, make these interfaces more error-prone, and thus entail the learning curve is slower, or make the work more tiring and thus have negative consequences for physical and mental well-being, and, in fact, may come to flow into all these areas.

Ergonomics within the Human-Computer Interface covers all the aspects of computer systems that influence the participation of users in their tasks. Therefore, Ergonomics is interested in both the utility (suitability for the task) and the usability (ease of use) of computer products and systems so as to seek to ensure the suitability of the software, particularly the interfaces, for the user's interaction tasks and goals. This corresponds, in practical terms, to the ability of the software to "enable" users to meet their goals (how to compose a text, to print, to use a spreadsheet, to browse in a hypertext) easily, effectively and with less physical and mental effort.

The interest in using ergonomic principles when designing and evaluating HCI stemmed from the need to avoid gross conceptual errors and to facilitate decision-making by designers and evaluators in order to save time and ensure the greatest consistency and uniformity possible. These principles have traditionally been translated into checklists developed by ergonomists who specialize in HCI, thus resulting in methodologies based on experts' views. Today, ergonomics has several methods available that can be applied in systems and software analysis. Soares [1] states that Ergonomics offers the researcher parameters for investigating physical and organizational matters, especially in industrial environments and the service industry where these are directly influenced by the behaviors, whether conscious or unconscious, of users who are found to concentrate on activities which involve decision taking, sometimes in situations where there is considerable tension and stress.

2.1 Design of Interaction

Design has been a strategy that has been increasingly used due to its importance as a factor that can make a difference, and add value and quality to products and services. The manifestation of Design occurs, above all, through two qualities: functionality and style.

These processes related to Design are directly linked to building a systemic environment that is suitable for use. Items of software constantly force users to perform tasks that could very often be reduced by up to half, according to studies in the area, as has already been shown. What is still needed is to analyze and design a he virtual environment focused on various matters present in the need for each task/project. According to Filatro [2], interaction does not happen by chance in activities undertaken, given that there is a computer system. It needs to be planned intentionally and to be expressed visually and functionally in the interface being used. Preece, Rogers and Sharp [3] state that it is the "Design of interactive products that provides support to people's everyday activities, whether at home or at work."

2.2 Usability

Within the multidisciplinary nature of Interaction Design, part of the process of developing an interface is about clarity when the artifact is being used. To obtain this clarity, some goals in developing the design must be achieved such as designing it so that it may offer efficiency in its use, thus enabling users to obtain high productivity when interacting with the artifact; showing good feedback to users; and providing support to effective learning. These goals are related to the usability of the artifact.

Usability is defined by ISO 9241 as “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.” The effectiveness with which the user will achieve his/her goals means he/she is able to accomplish the task successfully. The (less) time it takes to perform this task is about efficiency while satisfaction defines the extent to which the user finds the system acceptable [4]. It may also be defined as the degree to which the user performs a task [9].

Since the ISO definition consists of objective matters such as productivity in the interaction, and subjective ones such as the user's satisfaction with his/her experience with the artifact, it is seen to be very flexible. According to Preece, Rogers and Sharp [3], usability is considered as the factor that ensures that artifacts are easy to use, efficient and pleasant, from the user's perspective. Specifically, the authors divide the goals of usability into: (i) being effective in use (efficiency); (ii) being efficient in use (efficiency); (iii) being safe in use (safety); (iv) being of good utility (usefulness);

(v) being easy to learn (learnability); and (vi) being easy to remember how to use (memorability).

One of the best ways to identify usability problems of a known interface, or even an unknown one, is by evaluating it using prototypes, models, systems, etc. According to Prates and Barbosa [5], the main objective of evaluating an interface evaluation is to analyze the quality of its use of a piece of software or of a digital artifact. A big question is how to know what to assess when evaluating and defining the importance of each type of evaluation. These questions drive fundamental tasks within a good interaction design.

As stated by Harasim [6] one of the greatest problems in the advance of systems, technologies and computational tools lies in the fact of their very often being only in the hands of engineers and computer science professionals. Given this, what happens, in general, is that these professionals have no major concerns with (i) perception and (ii) usability, which are two fundamental aspects in an interaction process. Because of this, what can occur when designing projects are flaws in interactivity, iterability, logical consistency, and learning (in the broadest sense), thereby causing noises in the SHTM - System-Task-Human-Machine interaction.

Robertson [7] states that when an ergonomic analysis is structured via management tools, it is proposed at that moment that ergonomic solutions stop being isolated or taken one at a time and start to take on a macro scope (macro-ergonomics), thereby bringing out the potential of their results and contributing effectively to the well-being of the community. Similarly, following the same reasoning, it is possible to have in hand a diverse range of tools that can be used via Ergonomics so as to focus on improving activities related to HCI, and which will make a difference during the development process of the proposed methodology.

2.3 Ergonomic Evaluation Methodology for Software – Process and Tools

The focus of the architecture of information is on the structure of the system and not on its functionality or appearance (even though in practice they are linked). Thus, there is the objective of constructing easy-to-use systems/pieces of software that meet users' needs and objective. To achieve good usability in computerized IT systems it is important that the architecture of the information is clear, and this is obtained by arranging this information well, providing help to users regarding what they are looking for and differentiating between what is a priority and what is secondary wherever they may be working [8, 9]. According to Agner [10], there are five interdependent systems that form the architecture of information, each with own rules: a) the **system of organization** that determines how the organization is presented and the content is categorized; b) the **system for labeling** that defines the terminology and visual signs for each element of information that supports how the user browses; c) the **system for browsing** that specifies ways to move through the information space within the system/software; d) the **built-in browsing system** that is formed by the links to key areas of the site, access to subsections of the site and cross-references that link to pages of other sections with related themes; and e) **the system supplementary browsing** that is formed by the basic items of guides, indices, a map, search, etc. The advanced system, for its part, is formed by the options which allow search engines to be used in a personalized way.

In the process of evaluating usability, with regard to the activities of determining the usability and acceptability of the artifact, various criteria and requirements, which will determine the qualities of the system, must be taken into account, for example, the number of errors that users typically make when using software and computer systems.

According to Preece, Rogers and Sharp [3], assessing an artifact helps ensure that it will meet the user's needs. Cybis [4] backs this up further by adding that the assessment techniques of Ergonomics are diagnostics and are based on checks and inspections of the interfaces that look for interaction problems between the user and the system. According to these authors, there is a relationship between four models that can be used in evaluating interfaces: "quick" reviews; usability tests; field studies; and predictive assessment.

3 Developing the Project

3.1 The Problem Identified

All theory that involves usability works with the concept of "Heuristic Evaluation", which is nothing more than an empirical method for evaluating the usability of interfaces, in which generalized rules or guidelines are established, and which have been developed by experts with market experience in solving certain types of problems placed in particular areas of usability.

This method can be applied at any stage of developing the interface with the user (application screens), from prototyping to implementation. Furthermore, it is considered an easy, quick and cheap method of analyzing and diagnosing problems in interfaces.

The bibliography gives lists of heuristics with their respective descriptions but this is of little or no use for non-experts, or appraisers of software who may not have previously worked with usability analysis.

As there is no reference guide that gives pointers on how to apply and take decisions about the performance of the system analyzed in terms of usability, a non-expert appraiser will struggle to classify a system as compliant or non-compliant if it does not serve only all the heuristics but nevertheless serves others. Similarly, a non-expert would also have difficulty in choosing, from among the dozens of heuristics available, which ones are the most appropriate for systems analysis given the wide variety of computer platforms that are available today.

The study now proposed is there precisely to try to tackle these issues.

3.2 Process for Developing the Software

The language JAVA was chosen to develop the on-demand software of the contracting company. The modeling language was UML, the standard modeling language for developing projects that use object-oriented programming language [11].

The phases of the development process were: a) **Planning and drafting**, including Planning, Defining requirements and prototypes; b) **Constructing the system**, including coding and testing; and c) **Implementation**, involving the production for the end user's use, user training and maintenance of the system [10].

3.3.1. The Planning and Drafting Phase. This phase included: a) Standardization of Object-Relational Mapping; b) Architectural Design; c) Defining the initial conceptual model, including the Macro Flowchart of the Process; d) Prototyping; e) Surveying functional and nonfunctional requirements; and f) a Study that prioritized functionality and distribution between the iterations.

The system architecture is designed to be flexible when the application is evolving and to facilitate future maintenance, while at the same time being independent of database technology already on the market. Therefore, so that the architecture might achieve this goal, what the authors call a 3-tier architecture, or (3-Tier), was developed and comprises: a Presentation Layer; a Business Rules Layer; and a Persistence or Data Layer.

The elements belonging to the architecture were: a) the Entities (Basic Java Classes); b) the Repositories (database); c) the Controller (Business Rules); d) the Exception (Robustness); e) the Facade (Class that allows communication between the presentation and persistence layers) and f) the WEB (JSP, CSS, JS and Servlet).

3.3.2 The Construction Phase. Any one development process was iterative (by having multiple iterations in time) and incremental (by generating new incremented versions at each release). One of the advantages of this process is that the requirements change over time and an iterative process maintains frequent contact with the client which helps to keep the requirements synchronized. Allied to this, it is highly motivating for the development team (and the client) to see the software evolving at each release.

Thus, at each iteration there will be: a) Analysis (refining of requirements, refining of the conceptual model); b) Design (refinement of the architectural design, low level design); c) Implementation (coding and tests); d) Transition to product (documentation, installation, etc.; and e) Details on the Analysis of the Project.

The analysis generated a model to understand the domain of the problem. This stage of analysis also addressed, at a high level, how a possible solution can be assembled to meet the requirements from the user's point of view, and only dealing with the domain of the problem. UML diagrams were used to generate the modeling of the system and to facilitate the client's understanding.

3.3.3 The Design of the System. The design phase is an extension of the analysis model, now targeting the implementation of the software. The result obtained was useful for the programmer starting to develop Java classes.

The design activities included: a) Phase for refining the architecture (high-level design); b) Defining packages (modules), interfaces between packages; c) Decision-taking on the use/creation of libraries and/or components. Matters related to presenting information and the design of the interface were obtained from Tullis et al. [12] and Ashman et al. [13].

The system was divided into four modules which, in turn, are subdivided into smaller modules: a) an Evaluation Module; b) an Activities Module; c) a Heuristics Module; and d) a Results Module.

The detailed design phases (low-level design) were: a) Defining the attribution of responsibilities between the objects; b) Constructing diagrams of classes; c) Including javadoc documentation; d) Constructing interaction diagrams; e) Considering how to deal with flaws; f) Detailing the output format (user interface, reports); g) Defining the schemata of the Database (DB); and h) Object-Relational Mapping.

3.3.4 Implementation. Codes were written based on codification rules using good programming practices and following details described in the document for standardizing object-relational mapping. It is worth giving attention to the procedure of code reviews (review of source code), thereby seeking the good quality of the product developed.

3.3.5 Conducting Tests. Tests are fundamental to the quality of the final result. Although they have not yet been carried out, the conduct of two types of tests is laid down:

- Tests done by the programmer himself during programming, including Unit test (a test of individual classes or of groups of related classes), Functional test (a test of entire functions, e.g., menu item, Component test (a test of entire components (exe, dll, ...).
- Tests done by test teams, including a System test (which tests the integration of all the components of the product), Alpha test (a test of the entire product done by the development team), Beta test (a test of the entire product done by the Customer).

Finally, the transition phase, or final delivery of the product, will take place when the application hosted on the web application server specific to Java/JSP is made available. This step will involve: a) the production of a user's manual and b) the training of end users.

4 Conclusion

What was shown by having run the system proposed was that it is viable to encapsulate specialized knowledge on analyzing usability in a computer system to help non-experts conduct this analysis on other computer systems.

The application of the system in one of the electric power utilities in Brazil served to prove, in a pragmatic way, the positive effect of using the method in a real situation.

References

1. Soares, M.M.: Translating user needs into product design for the disabled: an ergonomic approach. Theoretical Issues in Ergonomics Science 13, 92–120 (2012)
2. Filatro, A.: Design instrucional na prática. Pearson Education do Brasil, São Paulo (2008)

3. Preece, J., Rogers, Y., Sharp, H.: Design de interação – Além da interação homem-computador. Bookman, Porto Alegre (2005)
4. Cybis, W., Betiol, A.H., Faust, R.: Ergonomia e Usabilidade, conhecimentos, métodos e aplicações. Novatec Editora Ltda, São Paulo (2007)
5. Prates, R.O., Barbosa, S.D.J.: Avaliação de interfaces de usuário: conceitos e métodos. Anais da Jornada de Atualização em Informática, XIX Congresso da Sociedade Brasileira de Computação, Campinas (2003)
6. Harasim, L., et al.: Redes de aprendizagem: um guia para ensino e aprendizagem on-line. Editora Senac São Paulo, São Paulo (2005)
7. Robertson, S.: Requirements trawling: techniques for discovering requirements. International Journal of Human Computer Studies 55, 405–421 (2001), doi:10.1006/ijhc.2001.0481.
8. Romani, R.: Usabilidade na Web, UNICAMP Campinas - SP. M.Sc. Dissertation (2006), ftp://ftp.unicamp.br/pub/apoio/treinamentos/tutoriais/tut_UsabilidadeWeb.pdf (Accessed in May 2012)
9. Brink, T., Gergle, D., Wood, S.D.: Usability for the web: designing web sites that work. Morgan Kaufmann Publishers, San Francisco (2002)
10. Agner, L.: Ergodesign e arquitetura de informação: Trabalhando com o usuário. Quartet, Rio de Janeiro (2009)
11. Rozanski, N., Woods, E.: Software Systems Architecture, p. 546. Ed. Addison Wesleyu (2005)
12. Tullis, T.S., Tranquada, F.J., Siegel, M.J.: Presentation of Information. In: Vu, K.L., Proctor, R.W. (eds.) Handbook of Human Factors in Web Design, 2nd edn., pp. 153–189. CRC Press, BocaRaton (2011)
13. Ashman, H., Dagger, D., Brailsford, T., Goulding, J., O'Sullivan, D., Schmakeit, J., Wade, V.: Human-Computer Interaction and the Web. In: Jacko, J. (ed.) The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, 3rd edn., pp. 565–587. CRC Press, Boca Raton (2012)