REVIEWING THE SACARWEB DESIGN BASED IN DISCOUNT USABILITY ENGINEERING

Fabrício Ferracioli
Universidade Estadual de Londrina
Mestrando em Ciências da Computação
fabricioferracioli@gmail.com

Maria Angélica de Oliveira Camargo-Brunetto
Universidade Estadual de Londrina
Doutora em Ciências da Computação
angelica@uel.br

ABSTRACT

This work analysed the SacarWeb's project and its objectives facing a usability analysis. The objective is to verify if the software attends its objetives of learnability, errors rate, if it is usable and to integrate usability techniques in the development cycle of the SacarWeb team. For this it was tested the Discount Usability Engineering approach, that proved to be adequate to teams with low usability knowledge, since it was found many valid usability problems and coincident among different evaluation types and evaluators. Our results have shown that is possible to a team make usability inspections based in the literature, that is possible to use the usability problems as a source of tasks to development team and that is not a difficult approach to learn, so it is possible to integrate in teams without usability specialists. Additionally we conclude that usability can't be neglected in the software development cycle, since many adoption problems can occur because of this kind of problem.

KEYWORDS: usability, web applications, discount usability, healthcare

REVISÃO DO PROJETO DO SACARWEB COM BASE EM ENGENHARIA DE USABILIDADE COM DESCONTO

RESUMO

Este trabalho analisou o projeto do SacarWeb e seus objetivos frente a uma análise de usabilidade. O objetivo foi verificar se o software atende seus objetivos de aprendizado, taxa de erros, se é usável além de integrar técnicas de usabilidade no ciclo de desenvolvimento da equipe do SacarWeb. Para isso foi testada a abordagem Engenharia de Usabilidade com Desconto, que provou ser adequada para equipes com baixo conhecimento de usabilidade, uma vez que foram encontrados vários problemas de usabilidade válidos e coincidentes entre diferentes tipos de avaliação e avaliadores. Nossos resultados mostraram ser possível a uma equipe realizar inspeções de usabilidade baseadas na literatura, que é possível utilizar os problemas de usabilidade como fonte de tarefas para a equipe de desenvolvimento e que não é uma abordagem de difícil aprendizado, sendo assim possível integrar em equipes sem especialistas de usabilidade. Adicionalmente concluímos que problemas de usabilidade não podem ser negligenciados no ciclo de desenvolvimento de software, uma vez que muitos problemas em sua adoção podem ocorrer devido a esse tipo de problema.

PALAVRAS-CHAVE: usabilidade, aplicações web, usabilidade com desconto, cuidados com saúde.

REVIEWING THE SACARWEB DESIGN BASED IN DISCOUNT USABILITY ENGINEERING

INTRODUCTION

Health care professionals at university hospital must be concerned to clinical issues as well as teaching and scientific ones. Frequently, clinical data can be a rich source of study and should be structured, in such way that it can be easily retrieved. Even though there are many alternatives of open source Electronic Patient Record (EPR), it is common that many health care units have difficulties to keep integrated the clinical data and results from a lot of exams, as well as to keep a temporal record of the patient's assessments performed. At scientific and teaching issues, it is desirable for the teacher/researcher not only a fast way to retrieve information, but also to get an explanation about each exam, including: how it is performed, what variables are collected, what variables are computed, details about computation, how to interpret results, and so on.

At Universidade Estadual de Londrina, there is a health care unit responsible for the physical rehabilitation of chronic respiratory diseases patients. Chronic diseases in general require continuous attention to control the occurrence of crisis. Aiming to help health care professionals (mainly in cardiac and respiratory rehabilitation) to accompany their patients individually and study groups of patients according to specific features, it was designed, developed and implemented the software SacarWeb (Software for Assessment of Cardiac and Respiratory System for Web). The assessment can be made through several kinds of exams, which have specific material and methods for both, execution techniques and analysis of results (interpretation).

One of the great challenges of SacarWeb is to construct a large database with shared clinical data from several health care units. The idea of sharing database of clinical data has emerged from the need to get a large set of cases of study for both issues (teaching and research). Such approach may be a fast way to reach the target, instead of considering the data of only one health care unit. It was though in a collaborative work, where each health care unit has a local administrator that is responsible for the data of their patients, with full access (reading/writing) on them, allowing (or no) to share their patients data with local administrators of other health care units (with permissions of data reading but with blinded names of patients).

The SacarWeb's design and development team consist of people related to computing and physiotherapy continuously working in the software. Despite of a lot of effort in producing improvements [1], the software has presented some difficulties of use, what can be explained by the lack of people with deep knowledge of usability. Aiming to reduce these difficulties and increase the trustful of the healthcare professionals, we have begun to investigate about SacarWeb's usability. Only common sense was used during the interface development and no one study or usability inspection was done in all the application life cycle, because of the lack of knowledge of this area in the members of the team.

Healthcare applications generally suffer of poor usability, resulting in low rates of software acceptance and high rates of software rejection [2, 3].

Intuitively the development team of SacarWeb begun to apply some fast usability tests with people from the team to make some design decisions based on what they think that is good about usability. Later, the team evolved to a better approach, trying to base their design solutions in the literature or in rational data about similar systems. That's the phase that the team is encountered today. Unsatisfied with design bad results, generic and poor data, the team decided to evolve to a better approach doing tests with target users and with well defined methodology. The evolution of care about usability from the team is like the one described in [4].

Despite of its importance, usability is known by its adoption difficulties in teams consisting only of software developers or people with deep knowledge in Computer Science [5]. Besides, it is not so common to include this discipline in the undergraduate Computer Science Curriculum [6]. Sometimes the professionals simply do not give the necessary importance to usability, thinking only in working software. Even considering those that think important to adopt usability techniques during the design, it is supposed that such techniques are difficult to learn and use, once they require high knowledge and experience [7].

Traditional approaches to evaluate usability are well known by their good results. Normally they give rich information about the use of the software and compliance with known patterns. These approaches have high time consuming, because of the amount of data generated by methods to analysis and require more knowledge to be performed, once the methodology of such methods is highly rigorous. One approach example is the full Usability Engineering life cycle, consisting of eleven general stages, where various options of methods are available, including traditional ones. Such approaches can discourage the adoption of usability techniques in teams without or little knowledge about them [7].

There are alternative approaches to traditional methods to evaluate software usability with similar quality of results indicated for people that know the importance of software usability, but do not have enough resources, or the necessary knowledge to adopt traditional methods [7]. One of these approaches is known by Discount Usability [8], that uses only the most important techniques of the full Usability Engineering.

Based on this, the objective of this work is to identify a set of usability problems in the core functions of SacarWeb, in a way to prioritize the future improvements in the application through Discount Usability Engineering. The system's core were presented in early papers [9], e [10].

2 linhas em branco

SACARWEB OVERVIEW

During the SacarWeb's design, some requirements needed to be attended, as follows:

(i) The software can be used from different locations, and should be easy to learn;
(ii) The software should attend different interests of health care professionals, including clinical and/or teaching/researching ones;

(iii) The software provides a large variety of exams, but should allow that health care professionals select only ones matching with their interests;

(iv) The software should provide information about the exams, including how the results have been reached, how the exam has been executed. It is important that health care professionals know how the software provides the results of the exams;

(v)Different health care unit databases can be shared for teaching/research issues;

(vi) The knowledge domain must be extensible (today it addresses cardiac and respiratory systems, but other systems are associated, as muscular system, nervous system, and others).

Based on these requirements from the physiotherapy team, the development team made the following decisions: the better way to provide wide access to the application is through a web based application (i), with different levels of access and permissions, attending the specific roles of each kind of user (ii).

Three kinds of users were identified: the local administrator, collaborator and knowledge expert. The local administrator is the person responsible for a health care unit (normally an experienced health care professional, including teachers and researchers) with permissions to to customize the system for their unit care, including the selection of exams of interest (iii) and to select the interface idiom, full access to the Electronic Patient Record (EPR), perform information retrieval using dynamical filters; see the evaluation of a specific patient according to graphic view of different exams and register their collaborators that will take the role of filling the EPR. The Evaluator user has permission to use only the Patient Assessment module (to register patients, introduce or view exams) and must report his actions to the local administrator. The Specialist has the role to add information about each exam, including details on execution procedure, variables considered, interpretation rules of the results, bibliography and so on. Such activities are related to the requirement (iv).

Future developments will be made to fully attend the requirements (v) and (vi).

The SacarWeb's architecture can be viewed in the figure 1:

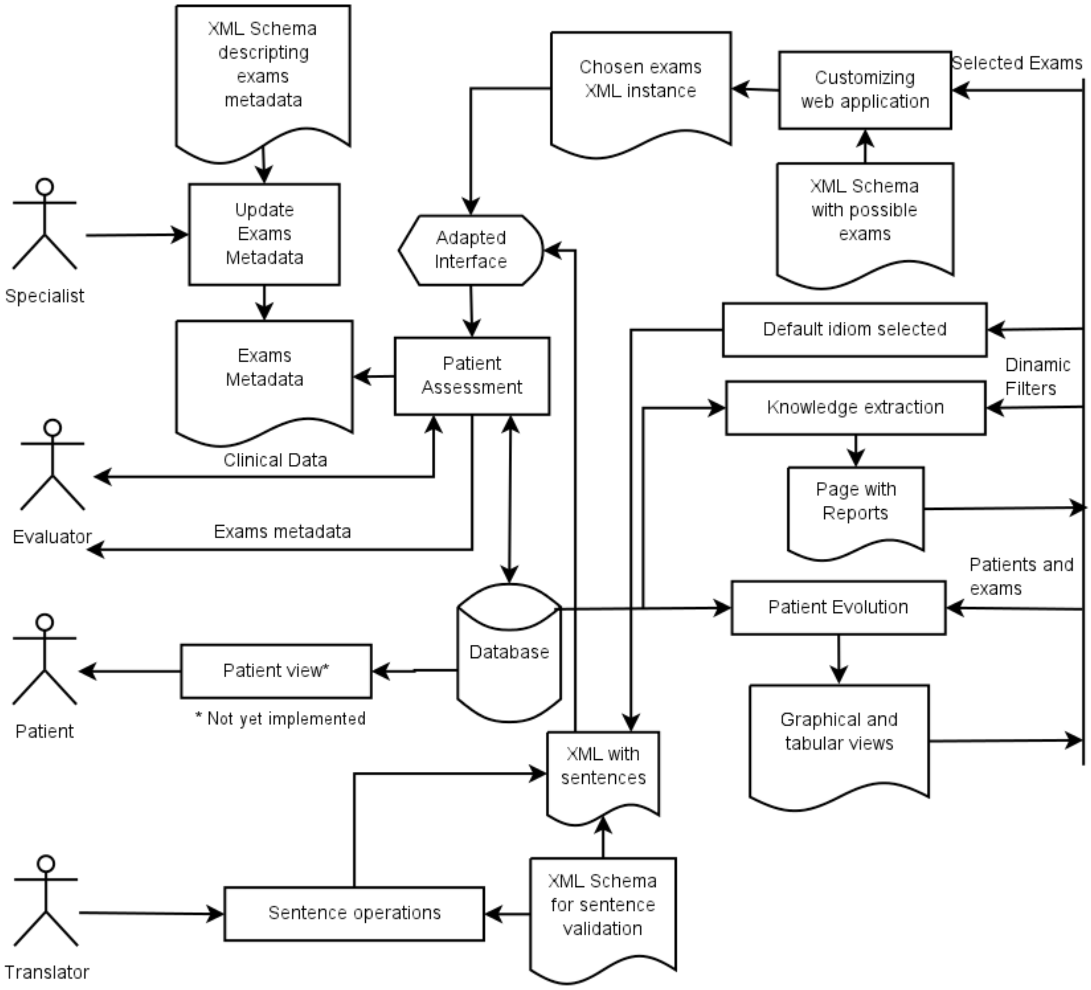

Figura 1: SacarWeb's Achitecture

By means the module Patient Assessment, which has the core system functionalities, it is possible to perform data collecting and analysis of patient exams. Each assessment is represented by a set of exams. The exams available are grouped by category: Questionnaires to assess quality of life: Chronic Respiratory Disease Questionnaire (CRQ) [11], adapted to Brazilian population, and St. George’s Respiratory Questionnaire Paul Jones (SGRQ) [12]; Tests of pulmonary function: The system provides the use of different equations of normal values, including Brazilian standard [13], American standard due to [14] and European standard by the Official Statement of the European Respiratory Society [15]; Exams to test exercise capacity: Shuttle Walk Test [16] and the 6 Minutes Walking Test [17].

An important feature of SacarWeb is the possibility to extract groups of subjects that satisfy different conditions, according to the interest of the health care professional [10]. There are some variables that are commonly used to filter data, as gender and age range. Besides these static filters, it was introduced dynamic filters that the user (local administrator) defines at execution time. A filter consists of a set of rules. Each rule includes: to select an exam, to select a variable of this exam and to select a relational operator specifying the range of values of select variable. The user can define so many rules so desire. This procedure will help the researcher to discover interesting features about different groups of patients.

During the application customization or the exam execution is important to the evaluator to have information about the exam, since its protocol, execution's pattern, formulas and information needed are key characteristics to the choose the exam. Different ways to interpret a result and execute an specific exam can be available, and the differences between them interfere in the results and interpretation. Because of it the module that describes the exams was made. The information of each exam is given by specialists in a collaborative editor and recorded as metadata about the exam. Since the information is given by specialists, their validity is higher. After this, these metadata is provided as a information about the exam or help to the evaluators.

The module customizing the web application allows that the local administrator selects what exams he/she intends to use [10]. The possible exams that the system provides are represented by a XML of the possible exams. When the local administrator selects his/her preferences, the system generates a XML document with an instance of exams of that health care unit. Another possible customization feature is the application's idiom, actually available in two different languages [18].

Various iterations for development of new functionalities occurred in the software life cycle, but no one tested the usability. Some of these functionalities were to improve the software use, others to improve the performance [1], [19], but the results were not the expected. Often the users claim for better consistency of the application, like the standardization of some application behaviors and screens, and to make the information about patients or other secondary functionalities easier to find and use.

DISCOUNT USABILITY ENGINEERING

The Usability Engineering defines quantitatively the usability goals of an application in the beginning of its life cycle. The results are taken throughout the product life cycle [5]. The full process has eleven stages, but an usability effort that does not include all the stages can be successful. This is the aim of the Discount Usability Engineering [7].

Discount Usability is an alternative to traditional Usability Engineering approaches that use only methods that require specific knowledge of usability, complex environments for the analysis and large experience. The purpose of Discount Usability is to identify usability problems even if the tests are made by people without experience or specific knowledge about usability. Because of this, there is a better chance to be used in practical design situations [7, 8].

The Discount Usability Engineering is based in four methods:

- User and task observation
- Scenarios
- Simplified think aloud
- Heuristic Evaluation

The four techniques are based in both inspection and test techniques. Inspection techniques do not use final users during the tests, test techniques use them. Better results are reached mixing these two kinds of techniques [20].

User evaluation is the basic principle and first step of any Usability Engineering method, and it is maintained in the Discount Usability. The aim of this task is to know the user, or at least to have a feeling on how the product will be used. The users observation must address their individual characteristics, the tasks performed, the reasons to make a task and the users' evolution. This can be achieved in many ways, like visiting the users' workplace, since the observation does not interfere in users' work flow. Because of it, the observer should keep quiet and let the users work as they normally do without interference [7].

A prototype is a working model built to develop and test design ideas that can be used to examine content, aesthetics and interaction techniques from the perspective of designers, clients and users. Usability professionals test prototypes as they perform typical tasks of the intended product use, once prototypes are a good way to find usability problems in early stages of the design process. Prototypes more similar to the final product are called high fidelity, while those less similar are low fidelity [21]. Scenarios are a kind of low level prototype that cut down complexity of implementation. They reduce the level of functionality and the number of features. They simulate only some parts of the application, but for that the user needs to follow a previously planned path [7].

In the simplified think aloud method one user at a time uses the system for a given set of tasks while is asked to think out loud. By verbalizing their thoughts, users allow an observer to determine not just what they are doing, but also why they are doing it. This is hard to obtain in other methods, and can be a rich source of information about the interface [20], pinpointing interface elements that cause misunderstandings. The main differences between traditional and simplified think aloud is the presence of psychologists, user interface experts and videotapes on the traditional method. To analyze these sources of information takes much time and can intimidate developers, for example. The simplified think aloud use only notes taken by the observer about what the user says and does while using the system [7]. A general protocol to perform this test is given by [22].

Heuristic evaluation is a systematic method for the usability inspection of one interface with aim to find problems in its design and it is considered the most common informal method [20]. The heuristics are collections of few usability guidelines that are good to be attended in an application. With them, even non experts can find many usability problems. The heuristic evaluation involves a small set of evaluators using the software and examining the interface, judging its compliance with recognized usability principles found in the heuristics. Its conducted verifying the interface and trying to formulate an opinion about what's good and bad about it. The findings can be completed with the results of the simplified think aloud [23, 24, 7].

As Discount Usability is an approach that aims to optimize the use of resources, it is easy to apply and can achieve results with better cost benefit than traditional methods. This is just because the emphasis is in early and fast interactions with frequent usability entrance.

METHODOLOGY

Preliminary results of this work were published in [25]. Only most common tasks used in SacarWeb were analyzed, because these are the most used in the software. The practical results inspecting the common tasks appear faster than analyzing the full software [7]. These tasks are:

- To insert patient’s identity;
- To insert patient’s exams results (six different exams);
- To verify patient’s evaluation through various evaluations;
- To apply a filter to extract group of patients satisfying certain conditions.

In Heuristic Evaluation, these tasks are analyzed by suitable evaluators that interact and evaluate the interface and in Usability Evaluation, these tasks are analyzed by evaluators too, but through observation of the users interaction with software. The data inserted in the software were collected from real evaluations with patients, and the printed forms were the same used by the clinical evaluators.

The aims of the usability test were:

- to improve the learning rate;
- to decrease the occurrence of user’s error.

These objectives were defined, considering that the most of the user are students of physiotherapy, with different grades of knowledge of the specific domain, and also there is a high rotating of users. It is important that the evaluation has clear objectives in order that the evaluator has the correct focus on users and interface during the evaluation [7].

Both categories of users during the evaluations: target users and heuristic evaluators use a similar computer environment, and it was established the following standard for the evaluations:

- Mouse and keyboard compatible device;
- Browser Mozilla Firefox version 3.6.x, with javascript and cookies enabled;
- 1024x768 minimum screen resolution;

In order to avoid external interference during evaluation, it was recommended the following: to minimize the presence or movement of other people in the room, no ringing phones, do not use other programs or operating system.

By default, Discount Usability recommends the use of scenarios to evaluate the software. In our case, the software is in advanced development stage. Because it is not the aim of this work to compare an alternative design, but to evaluate the actual solution, we decided to use the high fidelity prototypes instead of scenarios. The similarity with the final interface is the major characteristic of this prototype kind. By the use of the same methods, interaction techniques and look of the final product, high fidelity prototypes offer more realist interactions. Despite of this, any fidelity level is capable to identify usability problems, and there is no difference in the amount of problems found by the different levels [26,21]. The decision in this work was made mainly to speed up the evaluation.

APPLYING HEURISTIC EVALUATION IN SACARWEB

The heuristic evaluation was made in SacarWeb by three evaluators, where only one is part of the actual development team. This one has the best knowledge about the application domain among the evaluators, two evaluators have few or no experience with heuristic evaluation and one is an area specialist, which the results of the other evaluators can be compared. They followed the ten heuristics made by Jakob Nielsen [7], that can be applied in the majority of systems. In order to do the evaluation, one application guide was given to evaluators. This guide has information about the evaluation and system's knowledge domain. This information includes how to operate in specific tasks, how to read correctly the printed forms, which tasks have to be done, the tasks order and so on. One software maintainer was responsible for assisting evaluators during the heuristic evaluation, providing information about the knowledge domain or how to operate the application. This kind of information was given only if needed and after the evaluator have exposed the usability problem in question.

For a better understanding of the tasks to be evaluated in the software, heuristic evaluators were asked to perform an interface evaluation with high quantities of data to insert into the system. Each evaluator has introduced data of two patients with three complete sets of exams, in such way that each task was performed at least two times. During and after the evaluation the evaluators were asked to compare the user interface with the heuristics, and filling their findings in an online standardized document with the heuristics and tasks provided, in order to facilitate the application of the method of heuristic evaluation. For each task, the evaluator should only to identify the associated heuristic with the problem detected, to describe it and also to justify why such problem violates the heuristic.

After all the evaluations have been completed, each document with the findings of the evaluators was revised by the same SacarWeb maintainer that helped the evaluators, who compiled one document with all the usability problems found. This document was sent again to all evaluators for a severity measurement, that will guide the development of new solutions. The severity measurement was made using the proportion proposed by Jakob Nielsen [7], where:

- 0 – it is not a usability problem at all;
- 1 - cosmetic problem only, need not be fixed unless extra time is available in project;
- 2 - minor usability problem, fixing this should be given low priority;
- 3 - major usability problem, important to fix, so should be high priority;
- 4 - usability catastrophe, imperative to fix this before product can be released.

If the evaluators need to access the software to remember some interface problem and then rating the severity, they can use the same login and password used during the evaluation.

APPLYING USABILITY EVALUATION IN SACARWEB

The interface evaluation with users was conducted by one person of the SacarWeb development team, using the simplified thinking aloud method. In the tests, the evaluator has taken notes of all the user actions that provided some usability insight and his own considerations based in personal perception. At the same time, the user testing the interface was encouraged to share his thoughts and feelings while using the software. During the evaluation all the dialog was recorded to assist in the evaluation. After the user test, the evaluator wrote a document with his perceptions and usability problems found, based in the notes taken, audio recorded and his usability knowledge.

A pilot evaluation was made before the real evaluations with two not representative users. This evaluation was made to calibrate the real test. One change due to this pilot was the data to be considered for learnability and initial performance. Initially, the evaluation analysis the two interactions with exams sets inserted by the user. But the gain of efficiency from the first to the second interaction with interface is much huge, interfering in the point that we want to measure. It was expected that the same behavior occurred in the evaluation with real users. The pilot evaluation gave us a notion of time in what users can perform the request actions, what is important, to give an expected time to the evaluation for the real users during the tests.

The real evaluation was made with four potential users of the application, with no previous contact with the software, and basic knowledge related to application domain. The real users were divided by specialty into the system, testing only the exams they know. The number of users for testing is enough, once it is possible to find the majority of interface problems [7]. The tests were conducted only with the evaluator and user, at a room in the laboratory where it is common to use the SacarWeb.

RESULTS AND DISCUSSION

The Heuristic evaluations have identified many problems. The three evaluators were capable to evaluate the interface, based on the provided material. The amount of problems found varied with the knowledge of the application domain, where the greater the knowledge about the system's knowledge domain, better the results. The amount of problems found by each evaluator was [81, 32, 34], totalizing 147 problems, where 41 (27.89%) of them was common among all evaluators, resulting in 106 distinct problems found. The severity given by the evaluators to the problems encountered 11 problems (10.38%) with ratio 4, where 3 of them with a final ratio 4, and 8 with a real number as ratio with 4 as the nearest integer. This was an expected result, because a heuristic evaluation is better when several people conduct it, once more problems can be found [23]. Another factor that can influence the results are the major field of evaluator's knowledge. In this work, there were evaluators from Computer Science and Interface Design. The two kinds can found the same problems, but with different analysis and workarounds. Also, the problems related to specific knowledge domain are important, because they have great validity once they are based in the evaluator's knowledge about some specific topic that he deeply knows. The generated report was a useful resource to development team, because they can suggest solutions for future development and improvement of the system, based in well designed principles. The severity measurement can guide the development, where the problems with greater score will have priority in their solutions against others. Because of these results, it is suggested that the heuristic evaluation can be easily integrated in the SacarWeb's development cycle, and it is expected that when applied during the project the results can be better and the changes in the final software reduced.

During the tests with the real users of the application we found 107 problems, where 27 (25.23%) was considered serious problems. One important finding was that the general places of the system have common problems with different users. Some of them are related to problems with some concepts and terminologies used during the patients evaluation and into the system in different ways. Generally, the problems found aren't major problems, that interpose in the system learning, but some of them can severely slow the user's actions. Despite of these general findings, one major problem was found with one of the system's function, where 75% of the users could not complete the task. Another major finding was the lack of consistency in different parts of the system. Users created some bad and harmful habits for them during the system learning that made difficult the task completion, mostly because of unexpected system answers or behaviors, that were repeated through the entire system. In various places some behaviors were not necessary, but the users felt more comfortable when doing it, even when it slows down the process. One interest finding was the similarity of problems between the pilot and the real evaluation, but in some cases the real users have much better performance than the pilot users. It is suggested that this happened because of the knowledge of the domain that real users have. The performance gains were observed with real users, and also in the pilot evaluation, confirming the importance of the calibration with a pilot study. In the exams, the specific part of the system, we found that the user's knowledge of domain was important, and helped him to evaluate correctly the interface and to input the data, differing from the general users of the pilot evaluation. Despite some problems, and one task that is difficult to learn, the users felt comfortable using the system and they think that it can be useful in their daily tasks, speeding up and easing some process manually done.

Other important finding was the similarity of problems encountered in the heuristic evaluation and in the interface evaluation with users. With an amount of 29 common problems' (13.62%) between evaluations, the final quantity of distinct problems was 184. Some serious problems were found only in one of the evaluations, showing the importance of the two kinds used in this work. These problems were the ones that seriously impacted the users in the completion of tasks, terminology or concepts misunderstood that impacted in the comprehension of system or items that seriously slow down the users or stimulate them to redo a finished task.

About the two objectives of the usability tests, it is suggested that learning can be much improved after the interface redesign considering the serious problems found. As they are related to core concepts for the users, that curently are difficult to understand, a better interface approach can solve this. With this, it is expected that the amount of users' error decrease, because the software will be easier to use. Another three main problems are consistency, error prevention and simplicity of the dialogs. These three points are related to heuristics [7], and a better care of them in the some parts of the system can improve the user experience through the entire software.

With a better usability care and solving the problems found during this study, it is possible to accomplish better the requirements made during the SacarWeb's design phase. As a consequence of the dialogs complexity, inconsistency and high amount of user inducted errors, the software is not easy to learn which difficult the attendance of the different interests of health care professionals in the software.

The development team can use the findings of the tests to guide their next development iterations, having some clearly points with major problems to attack, and the severity measurement and user's feedback can guide the development. The maintainer that conducted the usability evaluations can guide the problems corrections, because of his knowledge about them, users and system. With this, it is suggested that is possible to integrate the Discount Usability in SacarWeb's life cycle and we have a set of usability problems to correct in their next development iteration. Despite of it, it is desirable that the usability problems can be transcripted in exact development tasks for the developers, giving for them a better source and a real development task to do, once the usability vocabulary is not clear for them.

A related study reported that evaluators, with similar knowledge of Computer Science and usability, as that observed in evaluation of Sacarweb, can differ from a good and a bad design after learn some usability concepts [8]. As expected by [7], people with little knowledge about usability can find usability problems with heuristics and conduct usability inspections. In our case, evaluators with more knowledge, or about the inspections methods, or about the system tested, obtained better results.

CONCLUSIONS

In this work, we presented the SacarWeb's project analysis based in a usability inspection using the Discount Usability Engineering. The results from the heuristic evaluation and usability analysis showed that a better care with the usability of the program is needed, since many problems that causes discomfort, difficulties and slow down the users were found. Some of these problems were related to application core concepts, what is a point to concern. With these results we think that is possible to a team without one usability specialist apply techniques based in the literature, which was our case. Finally we suggest a SacarWeb's redesign based in the problems found by the usability inspection transcribed in tasks and defined clearly for a better comprehension by the developers, what concludes the Usability Engineering integration with the development cycle.

REFERÊNCIAS BIBLIOGRÁFICAS

1. AMORIM, T. A., CAMARGO-BRUNETTO, M. A. de O., KASTER, D. dos S., FERRACIOLI, F. Remodelagem do software sacar-web usando técnicas de engenharia e reversa e reengenharia de software. Anais do CBIS 2006, 2006.
2. BADENOCH, D., TOMLIN, A. How electronic communication is changing health care: Usability is main barrier to effective electronic information systems. BMJ, 328(7455):1564+, Jun. 2004.
3. KUSHNIRUK, A. W., PATEL, V. L., CIMINO, J. J. Usability testing in medical informatics: cognitive approaches to evaluation of information systems and user interfaces. Proceedings: a conference of the American Medical Informatics Association / ... AMIA Annual Fall Symposium. AMIA Fall Symposium, p. 218–222, 1997.
4. NIELSEN, J. Corporate usability maturity: Stages 1-4. Nielsen Norman Group, 2006.
5. FERRE, X., JURISTA, N., WINDL, H., CONSTANTINE, L. Usability basics for software developers. IEEE Software, v.18, n.1, p. 22–29, Jan. 2001.
6. DOUGLAS, S., TREMAINE, M., LEVENTHAL, L., WILLS, C. E., MANARIS, B. Incorporating human-computer interaction into the undergraduate computer science curriculum. SIGCSE Bull., v.34, p. 211–212, Feb. 2002.
7. NIELSEN, J.. Usability Engineering. AP Professional, 1998.
8. NIELSEN, J. Guerrilla hci: Using discount usability engineering to penetrate the intimidation barrier. Nielsen Norman Group, 1994.
9. CAMARGO-BRUNETTO, M. A. de O., ROSSI, A. D. Uma aplicação Web para análise de exames cardiorrespiratórios. Anais do V Workshop de Informática Médica, 2005.

10. FERRACIOLI, F., CAMARGO-BRUNETTO, M. A. de O. A Web based application to help on assessment of respiratory system. Anais do XI Workshop de Informática Médica, 2011.
11. GUYATT, G. H. et al. A measure of quality of life for clinical trails in chronic lung disease. Thorax, v.42, n.10, p.773-778, 1987.
12. JONES, P.W., QUIRK, F.H., BAVEYSTOCK, C.M. The St. George's Respiratory Questionnaire. Respir Med. Sep, v.85(Suppl), p.25-31, 1991.
13. SBPT. I Consenso Brasileiro sobre Espirometria. Jornal de Pneumologia, v.22(Suppl), p.99-165, 1996.
14. KNUDSON, R.J. et al. The maximal expiratory flow-volume curve. Normal standards, variability and effects of age. Am Rev Respir Dis, v.113, n.5, p.587-130, 1976.
15. ERS. Standardized lung function testing: Official statement of the European Respiratory Society. European Respiratory Journal, v.16(Suppl), p.1-100, 1993.
16. PAYNE,G. E., SKEHAN, J. D. Shuttle walking test: a new approach for evaluating patients with pacemakers. Heart, v.75, n.4, p414-418, 1996.
17. TROOSTERS, T., GOSSELINK, R., DECRAMER, M. Six minute walking distance inhealthy elderly subjects. Eur Respir J., v.14, p.270-274, 1999.
18. FERRACIOLI, F., CAMARGO-BRUNETTO, M. A. de O. Internacionalização de aplicações Web: uma ferramenta para acelerar a tarefa de tradução em um abiente de aplicação médica. Journal of Health Informatics, v.2, n.2, 2010.
19. FERRACIOLI, F., CAMARGO-BRUNETTO, M. A. de O. A utilização das recomendações do W3C no SacarWeb. Anais do XVI EAIC, 2007.
20. HOLZINGER, A. Usability engineering methods for software developers. Commun. ACM, v.48, n.1, p.71–74, Jan. 2005.
21. WALKER, M., TAKAYAMA, L., LANDAY, J. A. High-fidelity or low-fidelity, paper or computer choosing attributes when testing web prototypes. Human Factors and Ergonomics Society Annual Meeting Proceedings, p.661–665, 2002.
22. BOREN, T. RAMEY, J. Thinking aloud: reconciling theory and practice. IEEE Transactions on Professional Communication, v.43, n.3, p.261–278, Sep. 2000.
23. NIELSEN, J. MOLICH, R. Heuristic evaluation of user interfaces. CHI ’90: Proceedings of the SIGCHI conference on Human factors in computing systems, p. 249–256, 1990.
24. NIELSEN, J. Finding usability problems through heuristic evaluation. Proceedings of the SIGCHI conference on Human factors in computing systems, CHI ’92, p. 373–380, 1992.
25. FERRACIOLI, F., CAMARGO-BRUNETTO, M. A. de O. Using and integrating Discount Usability Engineering in the life cycle of a healthcare web application. Proceedings of the Health Ambiant Information Systems Workshop, HamIS 2011, 2011.
26. BARANAUSKAS, C., PALANQUE, P., ABASCAL, J., BARBOSA, S. Human-Computer Interaction – INTERACT 2007, Lecture Notes in Computer Science, v. 4662, n.16, p.150–164, 2007.