Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Biomedical Informatics 38 (2005) 75–87

Journal of Biomedical Informatics

www.elsevier.com/locate/yjbin

Methodological Review

A user-centered framework for redesigning health care interfaces

Constance M. Johnson[a,*], Todd R. Johnson[b], Jiajie Zhang[b]

[a] *Department of Biostatistics and Applied Mathematics, The University of Texas, M. D. Anderson Cancer Center, 1515 Holcombe Boulevard, 447, Houston, TX 77030, USA*

[b] *The University of Texas Health Science, Center at Houston, School of Health Information Sciences, Houston, TX, USA*

Received 7 October 2004

Available online 30 November 2004

Abstract

Numerous health care systems are designed without consideration of user-centered design guidelines. Consequently, systems are created ad hoc, users are dissatisfied and often systems are abandoned. This is not only a waste of human resources, but economic resources as well. In order to salvage such systems, we have combined different methods from the area of computer science, cognitive science, psychology, and human–computer interaction to formulate a framework for guiding the redesign process. The paper provides a review of the different methods involved in this process and presents a life cycle of our redesign approach. Following the description of the methods, we present a case study, which shows a successfully applied example of the use of this framework. A comparison between the original and redesigned interfaces showed improvements in system usefulness, information quality, and interface quality.

Keywords: User-computer interface; Human engineering; Data display; User-centered design; Usability assessment

1. Introduction

The goal in the design of user-centered software is to create systems that are modeled after the characteristics and tasks of the users. Employing cardinal axioms of good design early and throughout the design life cycle gives rise to systems that are easy to learn, increase user productivity and satisfaction, increase user acceptance, decrease user errors, and decrease user training time. In converse, not doing so often requires the redesign of a system. Redesigning interfaces is not only time consuming, but costly and frustrating for both the users and designers.

Health care software developers often overlook relevant user characteristics, user tasks, user preferences, and usability issues, resulting in systems that decrease productivity or simply remain unusable. The US General Accounting Office, a major supporter of software engineering, found that 98% of software designed for the US government was "unusable as delivered" [1]. Sittig and Stead [2] additionally found this same problem in clinical information systems. In one system reviewed system implementation took three years longer than intended and the cost was approximately three times greater than the original budget [2]. Several factors could be ascribed to poor systems development such as cost and time restrictions and/or developer lack of user-centered design knowledge. Only 61% of information system projects meet the requirements of the customer's specifications [3]. Furthermore, 63% of projects go over their estimated budgets with the top cited reasons related to initial inadequate user analysis [4]. Simply put, not enough resources are being allocated to basic design principles, especially in the beginning phase of a project. Fixing a problem in the development phase is estimated to cost 10 times more than fixing a problem in the design phase. Fixing a problem after

* Corresponding author. Fax: +1 713 563 4242.
E-mail address: cmjohnson@mdanderson.org (C.M. Johnson).

1532-0464/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2004.11.005

shipping a system costs 100 times more than fixing a problem in the design phase [5]. Incorporating good design principles in the beginning phase of a project not only saves time and money, but also decreases design changes late in the development process [6].

In the current information age, health care providers are challenged with an increasing amount of information, and therefore have a greater need to utilize technologies to efficiently manage such information. Their ability to easily adopt and implement these technologies depends upon the ease of use of these tools. Information technology is changing the way patient information is obtained and gathered and can impact the decision-making processes of clinicians. It is well documented that poor information displays can lead to inefficient care which may include redundant ordering of tests or missing information important to the diagnosis of the patient [7–9]. The key is having the right information in the right place for the right clinician. Clinicians need a concise conceptualization and representation of complex clinical data for accurate problem solving and decision making. Therefore, health care applications must be carefully crafted, considering the different backgrounds and tasks of the health care providers, to ensure not only that these programs meet the standards and models outlined by the profession, but are intuitive and easy to use.

There are several valid user-centered design methodologies, however, none address the methods required in the process of redesign. This paper reviews the methods required in redesigning user-centered interfaces and presents a framework for the redesign process. These methods are based on well-published guidelines from the areas of computer science, cognitive science, psychology, and human–computer interaction. Furthermore, in the case study we show how we successfully applied this methodology to the redesign of a difficult-to-use health care system that in the end showed significant improvement in system usefulness, information quality, and interface quality.

2. Human–computer interface design methods

There are well-published guidelines and principles for designing systems that provide comprehensive utility or functionality and usability. The "Logical User-Centered Interactive Design Methodology" [10] and others [6,11] propose valid user-centered methodologies. Although their methods focus on the design methodology required to build user-centered software from the product conception stage to full roll-out of the software, our methods focus on redesigning deficient health care software that has been previously rolled-out.

Typically, once a software product has been conceived and the user population identified, the next step is to conduct the following analyses: a user analysis, an environmental analysis, a task analysis, a functional analysis, and a representational analysis. Each of these analyses provides different, but necessary components in order to design the initial prototype or redesign a flawed system. User analysis consists of examining the characteristics of the intended users. Environmental analysis not only examines the environment in which the users work, but also their social and cultural milieu. The task analysis considers the tasks and goals of the users. The functional analysis is a high-level process that focuses on structures of the work and the cognitive activities of the users. Finally, the representational analysis examines the optimum information display format for each task. These steps are usually taken during the initial design process.

In the redesign process, not only are these analyses considered, but comparative analyses, heuristic evaluations, comparisons of the users' and the designers' conceptual model, small-scale usability studies, and a comparison of the old and new system are also conducted to identify problems within the original system and uncover potential flaws within the redesigned system. The remainder of this section provides details on all the steps used in the redesign process.

2.1. User/environmental analysis

One of the most important issues in the design of usable applications is to learn about the people who will be using the application. This information is needed because different types of users require different types of interfaces [12]. Novice users, for example, will require frequent informative feedback as opposed to expert users who will require rapid response time and the availability of shortcuts [11,13]. A user analysis profiles the characteristics of the intended users of the system, such as age, education, skill level, cultural background, goals, computer literacy level, frequency of use, and familiarity with the domain. User capabilities can be determined through a survey/questionnaire approach, direct interview, or direct observation [14,15,11]. A major goal in the development of usable health care software should be to design a system that matches user's capabilities.

Environmental analysis specifies the conditions in which systems are used. Several aspects of the physical environment are significant to the design of the interface. The place and conditions in which the system is used can be a deciding determinant for the type of interaction the user has with the system and the user's progress with the system [16]. Examining environmental issues such as space, lighting, noise, availability of resources, danger, speed, power sources, and social and cultural issues are an integral part of the environmental analysis [15].

The social environment of the users can impact the success or failure of a system. Social issues that need to be addressed as part of an environmental analysis are: (1) Do the users share information and work together; and (2) Are resources readily available to assist the users [15]. Cultural issues are significant to consider and not only relate to ethnicity, but also to socioeconomic status, professional status, and regional differences. These differences can affect working routines, interactions with others, values, and biases [15]. Taking the characteristics of the users and the environment into account addresses only two points of the triad of good human–computer interaction design. The users' tasks must also be considered using a technique called task analysis. We must also consider the users' tasks by using a technique called task analysis.

2.2. Task analysis

Task analysis is the process of identifying system functions that have to be performed, the required input and output formats, system constraints, information categories and flow, and the communication needs of the users [12,15,17,18]. Essentially, it examines the goals of the users and how the users will or should interact with the system to reach these goals.

There are many different types of task analyses. Techniques such as questionnaires/surveys, interviews, observation, and laboratory field studies provide different types of data for analyses. One of the best approaches to obtaining large amounts of data concerning the users' tasks is field observation. Field observation is an ethnographic approach that involves observing users in their own environment [17,19,20]. The benefit of this approach involves obtaining detailed information about people performing their tasks in their natural environment over a sustained time period. This adds richness to the data that could not be obtained through a survey or questionnaire [21]. The limitation of this approach is the processing of the field notes, which can potentially conceal the complexity of the task observed [21]. Once analyses of the tasks have occurred it is necessary to consider how these data will be presented. Some data presentation techniques to consider are hierarchical analyses, flow diagrams, archetypes, scenarios, and sequence diagrams and tables [17,19,21]. How these data are presented will depend upon the task of the user and the most effective way to represent this information to the designers.

A task analysis should be considered an iterative process and one that is done at many different stages in the design process. The type of task analysis considered is mainly dependent upon the design stage and type of task. For example, in the health care environment, task analysis requires examining the roles of users from many different types of disciplines in many different types of environments. A task analysis should ensure that only the necessary and sufficient task features that match the users' capacities and are required by the task are included in the system implementation. Extra features that are not required by the task will only complicate the interface and generate extra processing demands for the user, thus making the system harder to use. It is a systematic approach to understanding a user's job and what the user is trying to accomplish [17,22,15,18,23]. Ultimately, it will guide the overall system design, the usability specifications, and system functionally. There are many types of task analyses and understanding the user and environment can assist in choosing the one that will provide the most useful information.

2.3. Representational analysis

Representational analysis identifies the optimum way to display information to the user according to their respective tasks. This analysis helps match each information display to each user task. One of the key concerns with representational analysis is to identify ways to represent information that decreases the amount of internal processing that goes into solving a problem. Different representations of the same abstract process can make the problem more difficult or easier [24]. For example, Zhang and Norman [24] describe carrying out simple multiplication with Arabic versus roman numerals. Although both numbering systems represent the same abstract process, it is difficult for people at ease with a decimal system to render a product using roman numerals. The same concept can be applied to the displays of information. A good representation of information should enhance problem solving by decreasing cognitive demands. In the case of an extended family history, it is easier to read and evaluate a pedigree (family tree) than pages of historical text. The availability of data alone without a graphic display of information forces the collection, maintenance, and integration of these data mentally, which increases the probability of error [25]. In addition, performance is improved when information is displayed extrinsically through pattern recognition rather than intrinsically through straight text which requires intense cognitive activities such as memory and deduction [25–27].

Representational analysis is an approach that breaks down a task into parts, examines the representational attributes of each part, and identifies the external representation that supports the task of the user [24]. The goal in identifying the external representation that best supports the users' tasks is to reduce the cognitive load of the user and increase the ability to easily reason in complex domains.

2.3.1. Types of interface designs

There are two basic interface designs; direct manipulation where the user is engaged with the objects of interest and the conversational system where the user is engaged in a conversation with the system using language as the medium of exchange [28]. A direct manipulation interface is an interface that provides a visual representation of the required tasks through the use of objects or icons. Although there are advantages to both, direct manipulation interfaces can be modeled after the users' real-world tasks. These interfaces use a natural representation of task objects and actions that mimics the users' tasks. The benefits of utilizing such an interface based on metaphors are easy learnability, decreased human errors, and easy reversal of actions [11]. In general, the direct manipulation interface promotes exploratory learning, making the users feel more in control. The disadvantages are technical problems related to screen space and system resources. This interface causes problems if the users do not have the level of domain knowledge that is necessary to carry out the respective tasks [11].

2.4. Functional analysis

A functional analysis examines the relationships of the entities within the domain, the user goals and how the users will reach these goals, the structures needed for successful goal completion, and the information flow within the system [18,29]. There are two main parts to the functional analysis: (1) the work domain analysis and (2) the cognitive work analysis. The work domain analysis considers the structures of the work domain. The cognitive work analysis as it implies, examines the cognitive activities in the work domain. This type of analysis is an abstract process that identifies top-level domain structures.

3. Human–computer interface evaluation methods

3.1. Inspection methods

The objective of inspection methods is to uncover problems with user interfaces and make suggestions for fixing the problem [30]. They are considered evaluation methods that are completed when the user interface is ready for user testing. There are published methods for specifically evaluating clinical information systems [30a]. To begin, we describe one of the most common inspection methods—heuristic evaluation.

3.1.1. Heuristic evaluation

Heuristic evaluation is one of the most commonly used inspection techniques due to its low cost and low skill requirements [31]. The technique requires that a small set of experts evaluate a user interface based on their knowledge of human cognition and interface design rules of thumb (heuristics) [11,12,32,33]. Once the violations of the heuristics are identified, experts rate the problems in terms of severity on a scale from 1, indicating a cosmetic problem (fix can wait), to 4, indicating a catastrophic problem (immediate fix) [32].

Heuristic evaluations are generally good at exposing the majority of usability problems within an interface [34]. However, heuristic evaluations cannot reveal all problems, and the strength of the test lies in uncovering local problems. Using this technique along with other techniques will reveal both local and global problems [35].

3.2. Comparison of the users' and designers' conceptual model

The users' and designers' conceptual model or mental model of a task as defined in the functionality and usability of an interface is sometimes different. The designers' mental model of the users' tasks are formed through the task analysis and are represented in their design of the interface. The users' mental models of their tasks are formed through their interaction with the system. Mismatches between the users' and designers' respective mental models causes problems on both sides. For the user, there are problems with easily getting what they need from the system and not having an open dialogue with the system. For the designer, mismatches increases their workload through increased technical support or worse, redesign of the entire application. We discuss several evaluation techniques that disclose mismatches before the application is rolled-out.

3.2.1. Keystroke level model

Comparing the users' and designers' conceptual model using the technique of the keystroke level model can identify temporal problems within an interface [36]. The keystroke level model shows differences in execution times of each performed task by summing up the time taken for keystrokes, pointing, clicking, thinking, waiting, and deciding. Although the keystroke level model is tedious to perform, it can show problems with the predicted execution times of an application and point out particular areas where a user might be spending an inordinate amount of unnecessary time with the functionality of an application. It is a good method to use where time is a factor in performing tasks.

3.3. Cognitive walkthrough

Comparing the users' and designers' conceptual model using the technique of the cognitive walkthrough can identify learning problems within an interface [30b]. The cognitive walkthrough can disclose many problems that a first-time user would encounter with system functionality and ease of system use. It defines how well the interface supports "exploratory learning," or how well the first time user can perform a task without formal training [30b]. It is a technique that focuses on errors in design that would interfere with the users performing a task. It also explains mismatches between the users' and the designers' conception of a task [37]. The cognitive walkthrough consists of answering a set of questions that identify the users' goals and how easy it is for the user to meet these goals. Before beginning this type of analysis, the designer must know their users, the respective tasks they will be performing, and the accurate action order for each task [37].

3.4. Comparative analysis

Comparative analysis examines different aspects of other similar commercially available applications. This type of analysis compares the original applications with the other application costs, program type, functionality, usability aspects of the interface screens, import/export functions, and user platforms. It assists with defining alternative representations and is a significant part of the analysis in redesigning interfaces [12]. It can provide design ideas and determine what is good and what is bad about the existing products on the market.

3.5. Small-scale usability studies

Small-scale usability studies are an important way to validate interface design decisions and to test alternative interfaces. These studies include talk aloud methods [30] in the controlled environment of the lab, in which the users talk about what they are doing and thinking out loud as they use the interface. The aim of this technique is to collect procedural information about mental processing; wherein the investigator can make deductions about problems a potential user may have with an interface. The use of audio-video recordings while the subjects are working with the interface provides a rich source of data for later coding and analysis [38]. These types of tests uncover hidden functional and interface design flaws. Specifically, the studies uncover differences between the users' and designers' mental models of an application.

3.6. Comparison of the redesigned application with the original application

Comparison of a newly designed application with the original application in a controlled laboratory environment can show whether the redesigned application is better, worse or unchanged in terms of functionality and usability. This type of test is constructed as an experimental design where subjects are randomized and there are dependent and independent variables. One method to carry out this type of study is to first determine a set of typical tasks performed within both the redesigned and original interfaces. Once a representative sample of subjects is recruited, they are randomized to first carry out these sets of tasks in either the original or redesigned versions. The subjects are asked to talk aloud while performing these tasks. While the analyst takes field notes, the voices of the subjects are audiotaped, while the computer screens are videotaped for later analysis. After the subjects have completed the tasks in one application, they are asked to complete a questionnaire measuring overall user satisfaction regarding the ease of use, user efficiency, productivity, error rate, and many other outcome evaluations, before carrying out the prescribed set of tasks in the other application. There are several such questionnaires that have been tested for reliability and validity [39–41]. The results of the questionnaire, task completion time, and task success are measured and compared to determine if the redesigned application shows improvement over the original application.

4. An approach to redesigning a health care application: a case study

In this section we provide a case study based on our prior experience in redesigning a health care application using a redesign approach that combines several design and evaluation methods. The order of the steps presented herein is the sequence that should be considered in redesigning applications. The various approaches described above are shown in our case study below.

In 1997, a family history-tracking and pedigree drawing program was designed at a large teaching hospital in Houston, Texas for conducting genetics studies as a part of an academic program [42]. Although this tracking program had much functionality, an initial user survey and usability analysis revealed important missing functions and a host of usability problems. The tracking program needed more externalization of information and perceptual cues for operating procedures so as to increase the directness of the interface. Without good functionality and usability, it remained limited in its utility in clinical, research, and educational settings. Thus, we began conducting various analyses to determine the extent of the functional and usability issues. The results presented herein show how the approach we used can be used in the redesign of any type of software. These results show how we used the methods previously outlined to successfully redesign a health care application.

4.1. Step 1: Analysis of the original application

To establish what components of the original system needed to be redesigned, we first conducted a user, environmental and task analyses, heuristic evaluation, and a comparison of the users' and designers' conceptualization of the tasks. These analyses provided the empirical evidence for the process of redesign.

Initially current users of the system were categorized along a horizontal dimension, according to their different types of tasks, and a vertical dimension, according to their different levels of experience for specific types of tasks using a matrix. Along the horizontal dimension, existing users of the tracking program were health care professionals. Along the vertical dimension, existing users were computer literate at the novice, intermediate and expert levels, males and females with education levels ranging from high school to post graduate degrees.

Although this was a rather homogeneous population in terms of computer knowledge, most problems arose with this population when the tracking program was not consistent with other programs they had used in the past. In this regard, there were user errors, complaints, and loss of productivity. Open-ended interviews with the users provided additional information on what the users liked and disliked about the program and what extra features they would like to see incorporated into the program.

The environmental analysis showed that this application was used mainly in the private office environment without potential exposure to patients and other individuals. The system was only available to authorized users and was password secured.

The initial task analysis consisted of open-ended interviews to determine the users' tasks, including what, how, and when the tasks were done. These data were reviewed with the users through scenarios, tables of tasks, and a hierarchical task analysis.

The tasks of the current users were analyzed by a hierarchical task analysis, which was completed on each task in the original application. This type of task analysis allowed us to determine the goals of the users, the required functionality of the redesigned program, and it additionally identified steps in the original program that imposed an unnecessary cognitive load for the user. For example, to print a pedigree in the original program, the user had to perform a total of 29 steps, which included finding the family to print the pedigree. It was determined through this methodology that tasks such as this could be significantly reduced in the redesigned program.

The tracking program was thoroughly assessed for problematic areas through the use of several approaches. Number one, a heuristic evaluation uncovered at least one heuristic violation in all of the categories reviewed [11,12]. There were problems with visibility, consistency, use of natural language, informative feedback, minimizing the users' memory load, reversible actions, error messages, flexibility, and many other violations [43]. Although these usability problems have been previously reported, one of the catastrophic problems created by the designer was the use of cryptic, programmer-defined variable names in the user interface screens instead of changing the names to reflect natural language. This is not only problematic for first time users, but all users. An example of this can be seen in Fig. 1.

A comparison of a first-time users' and designers' conceptualization of each task showed that the users' mental model acquired through their contact with the system did not mirror the designers' conceptual model displayed through the system's model. A breach between these models resulted in major usability and functional problems. For example, the buttons shown in Fig. 1 were confusing to first time and seasoned users. If the

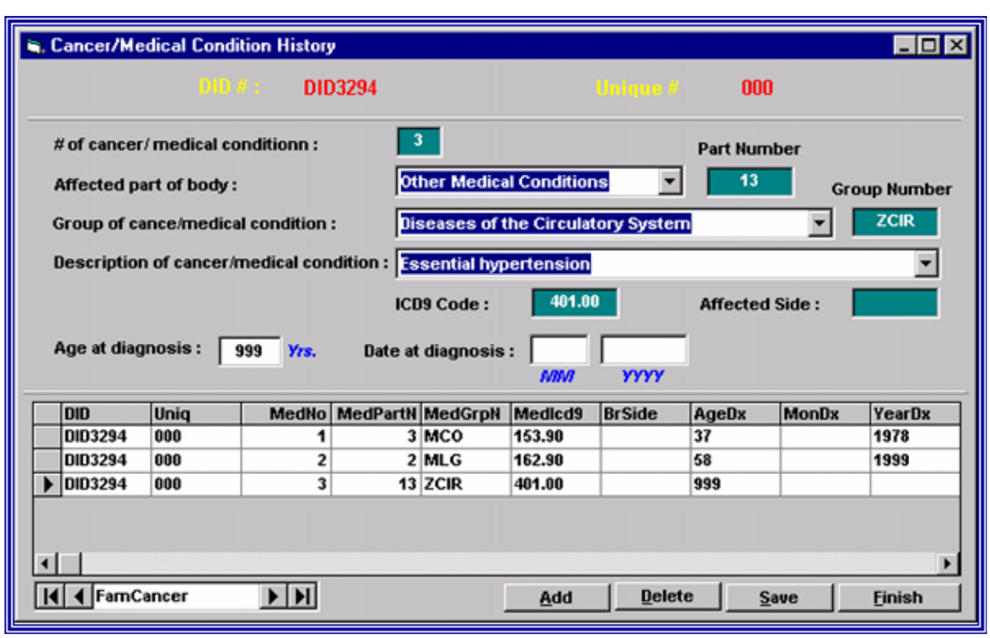

Fig. 1. Data-entry screen in original application.

user finished entering all medical conditions on a particular patient and did not click on "save" before clicking on "finish," then these data would not be saved and the user was not warned about this action. Often the user did not know these data were not saved until they printed a pedigree and these data were missing. This required a significant amount of extra steps. A key-stroke level model showed a reduction in the number of key-strokes if the program provided a simple error prevention message.

It was determined that the application needed more externalization of information that would increase the directness of the interface. It needed a reduction in the amount of effort that a user must exert to interpret the physical state of the system. Significant changes would be required to minimize the users' memory load, increase the speed of mental operations, decrease learning time, and increase the users' control of the system. The next step was to define the characteristics of the potential users of such an application to generalize the application.

4.2. Step 2: User analysis for the redesigned application

To determine the needs of the potential users of a family history tracking and pedigree drawing program, a direct mail survey was sent to 1252 full and associate US members of the National Society of Genetic Counselors (NSGC). Survey items were mainly closed-ended with many dichotomous and mutually exclusive items. Survey items included demographic information, computer usage and experience, and use of technology in the collection, storage, and usage of family history information. The demographic section included the respondent's age, gender, level of education, native language, occupation(s), and years of experience in each occupation. The family history information collection, storage, and usage section included how the family history information is currently collected, where it is stored, and how it is used. The computer usage and experience section was comprised of access to a computer at work, frequency of computer use, type of operating system used, and familiarity with different types of software.

A total of 481 surveys were returned, with a response rate of 38.4%. The results from this survey demonstrated that user analysis through direct-mail questionnaires is a proficient way to obtain information about potential users and to understand their needs. This survey enabled us to determine important characteristics of the intended user population and directly affected the redesign of the family history tracking and pedigree drawing interfaces. For example, since the respondents reported that drawing a pedigree freehand is the most frequently used collection method, we decided that a direct manipulation interface would best mimic this process. Since one third of the respondents reported that their work environment is distractive, it was decided to design an interface that supports easy resumption of tasks.

While approximately 30% of the respondents reported using the computer to collect family history information, and close to 40% responded that they store this information electronically, nearly 98% have access to a computer at work and over 95% use a computer every day. Based on this survey, we determined that nearly all respondents could potentially use specialized software to increase their productivity.

4.3. Step 3: Comparative analyses

The program was then compared to three other commercial data management and pedigree drawing programs. The comparison included program cost, program type, functionality and usability aspects of the interface screens, use of taxonomies, import/export functions, user platforms, and availability of analysis tools. A complete hierarchical task analysis and data entry time comparisons were completed on the commercial pedigree drawing programs as well, and compared to the original version. These analyses allowed us to make a comparison of different programs at a very granular level.

This comparison showed that there were no significant differences in the cost between any of the programs. All of the programs had a relational database component; however, in just two of the programs could the data be directly and easily manipulated in the backend database. Only the original family history application offered immediate use upon opening of the program the first time and a built-in taxonomy for coding medical problems. This analysis showed that additional security features, import/export functions, ability to interface with other analysis tools, and the addition of a direct manipulation interface needed to be added to the redesigned program.

To address the issue of the hybrid direct manipulation and form-fillin versus strictly form-fillin interfaces, data entry time was compared to determine if the object-oriented nature of one commercial program permitted faster data entry than the strictly form-fillin method of original family history application. Two experienced users entered data on a total of 10 families into the original family history application and a commercial program that offered a direct manipulation interface along with form-fillin. The results showed that although the commercial application was on the average a faster method to enter data, the difference was not significant. It took an average of 49 min to enter an average sized family (40 members) in the direct manipulation and form-fillin commercial application versus 52 min in the form-fillin application. The major difference noted was in the amount of editing time that was required in the form-fillin versus the direct manipulation and form-fillin application. On average the editing time in the direct manipulation and form-fillin application was 2.6 min per family entered versus 10.5 min per family entered in the form-fillin application. This difference occurred because the form-fillin application forced the user to enter all data, print or preview the pedigree, and then go back to the application to fix problems instead of fixing the problems as data were entered. This suggested that a combination of data entry and automatic pedigree drawing at the same time would offer the most timesavings.

Finally, the comparative analysis assisted with defining alternative data display representations. A hierarchical task analysis of both the original family history application and another commercial application was completed. The results of this analysis showed that the direct manipulation and form-fillin commercial application took significantly less steps to accomplish the same goals of the strictly form-fillin application.

4.4. Step 4: Functional analysis

The functional analysis connected each redesigned interface element with the overall structure of the application, what the element or function accomplishes when active, what it performs, and how the user interacts with the element. For example, the interface element, "Initial Dialog," is shown when a database within the application is not linked with the application. This dialog asks the user whether they wish to create a new database. The user interacts with the dialog by choosing an option and pressing OK or cancels execution of the application. This is only one aspect of the functional analysis and it shows in concrete terms the results of an abstract task. The redesigned process required an entire redevelopment of the structure of the application. In addition to providing the details of the functional elements, it also provided the information to change the overall environment to one that mimics the Windows Explorer type of view. The functional analysis, which involved careful consideration of the work domain and the cognitive work of the users, provided the structure of the redesigned version.

4.5. Step 5: Representational analysis

The presentation interface of direct manipulation and form-fillin was based upon the survey results of NSGC members, the comparative analysis, and the user information requirements discovered in task analysis. Although a combination of direct manipulation and form-fillin was considered from the beginning, the fact that the majority of NSGC members surveyed still hand-drew their pedigrees, even though they had access to technology, convinced us that we had to try to mimic that process through direct manipulation. The time saved and increased efficiency found upon comparison of a strictly form-fillin application versus direct manipulation and form-fillin in the comparative analysis further confirmed this choice.

4.6. Step 6: Creation of the prototype

We used the results of the analyses to build paper prototypes of the redesigned application. During this phase, high level navigation, design presentation, and the functionality of the displays were outlined. The initial prototype included data entry screens and the pedigree drawing component which allowed the users to draw the family and enter individual data and provided automatic data entry of some data linked to the pedigree component. The main page mimicked a Windows Explorer type environment to provide the user with ease of navigation and system state visibility. In this environment, the user could draw the pedigree through buttons offered or right clicking on an object on the screen. Dynasty Technologies (Houston, Texas) was used a consultant for the pedigree drawing component. The component for the pedigree drawing used C# as the implementation language and Microsoft Visual Basic.NET as the host application. The other features included automatic drawing directly linked to the data entry interfaces, customized shading, standardized pedigree nomenclature, customized subtext labels, full printing, and editing functionality. See Fig. 2.

4.7. Step 7: Small-scale usability studies

Small-scale usability studies are an important way to validate interface design decisions and to test alternative interfaces. Before conducting the small-scale studies, we conducted iterative heuristic evaluations of the system based on well-established usability principles and guidelines [11,12]. The prototype was revised until all major and catastrophic usability problems were corrected.

After the heuristic evaluations were completed, a small-scale usability test was conducted using talk aloud methods [30] in a controlled laboratory environment. A total of eight subjects were recruited and asked to complete twelve common tasks within the interface while talking aloud. The major problems identified such as how to begin data entry, continue data entry and label information on the pedigree were found by 50–75% of the subjects. These problems were addressed and corrected by modifying the interface to make the data entry and labeling more intuitive. In general, this usability test uncovered functional and interface design flaws.

4.8. Step 8: Modify the prototype

The results of the evaluation tests were used to modify the redesigned application. These tests also allowed us to identify further functionality that needed to be added to the interface. Once these problems were cor-

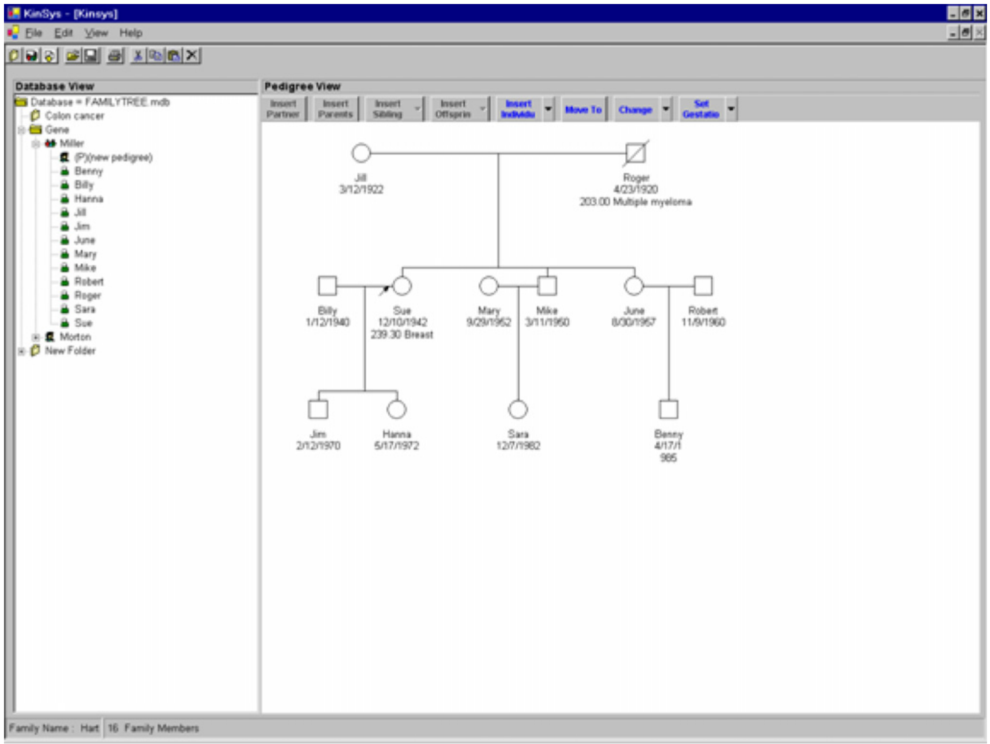

Fig. 2. Prototype of family history and pedigree drawing application.

rected, the program was then ready to be compared to the original version.

4.9. Step 9: Compare new and old systems

Finally, the redesigned program was compared to the original program using a controlled experiment to determine if the redesign decreased the error rate, increased productivity, and increased user satisfaction ratings. A convenience sample of 16 subjects was recruited to participate under an IRB approved protocol. After informed consent was obtained, the experimenter read to the subjects a scripted overview of the study. The subjects were informed that they would be carrying out a set of tasks in two different versions of a family history and pedigree drawing program.

To prevent learning bias, half of the subjects were randomized to use the original version first and the other half were randomized to use the redesigned version first. Subjects were given a set of written instructions for each version of the program. They were asked to carry out a series of 12 typical tasks in the old and revised versions of the program. The tasks included in the study were selected since they represented the most frequently performed functions in each version of the application. Although the task sets had minor differences due to the differences in the programs, they were considered functionally equivalent.

At the completion of the assigned tasks on each program, subjects were asked to complete the Computer System Usability Questionnaire for each version of the program [39,44]. The Computer System Usability Questionnaire is an 18 item questionnaire on a Likert scale that measures overall user satisfaction, but specifically "ease of use, ease of learning, simplicity, effectiveness, information, and user interface," p. 66 [39]. Administration and scoring of the questionnaire was followed according to the author's instructions.

4.9.1. Task completion time

A t test showed a significant difference ($p < .001$) between the times to complete the tasks in the original version and the redesigned version. The mean time the subjects took to complete all of the tasks in the original program was 48.3 ± 8.2 min with a range of 37–60 min compared to 34 ± 6.7 min in the redesigned version with a range of 24–46 min. This statistically significant time difference as in the comparative analysis confirmed the choice of the direct manipulation and form-fillin interface for the redesigned application.

4.9.2. Task success

Successful completion of each task by the subjects was compared within each version of the system. Successful completion of a task was determined by the user being able to start the task successfully within three minutes without help from the experimenter.

Chi-square showed that overall differences in successful user task completions between the original and redesigned versions were not statistically significant ($p > .25$). However, there were some differences in the completion of individual tasks within each version. Of the 16 subjects, 13 (81%) did not experience any problems in the redesigned program with opening the application and finding the data entry screen, but only 3 (19%) were able to successfully complete this task in the original version. There were two main problems with the original application: (1) opening the application required an inconsistent extra step (pressing the "Enter" key), and (2) lack of an intuitive menu name for the data entry screen. Although the subjects had some initial difficulty with figuring out how to enter the first family member in the redesigned system, 63% did not have any further problems. However, in the original version 56% ($n = 9$) continued to have problems with data entry after entering the initial family member. The original version had numerous usability problems which have been previously described [43].

4.9.3. User satisfaction

A Wilcoxon signed-ranks test was used to compare the differences between the original version and the redesigned version in terms of user satisfaction via the Computer System Usability Questionnaire [39]. First, general overall satisfaction was compared. Then the subscales of system usefulness, information quality, and interface quality were compared. Included in the system usefulness were questions relating to overall satisfaction, ease of use, efficiency, productivity, and ease of learning. Included in information quality were questions relating to clear error messages, recoverability from errors, system visibility, organization of information, ease of finding information, and ease of understanding. Finally, the questions relating to the interface quality included system functionality and in general, likeability of the interface.

The Wilcoxon signed-ranks test showed a significant difference with user satisfaction between the original and the redesigned version. The subjects rated the system on a Likert scale from strongly disagree (1) to strongly agree (5) with 3 as neutral on a series of 19 questions.

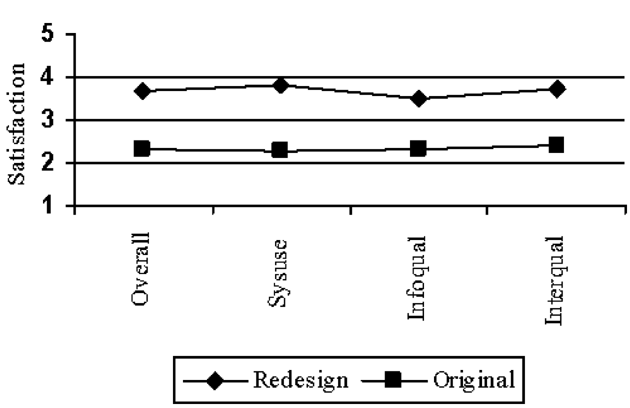

Fig. 3. Differences in user satisfaction.

Fig. 3 shows the mean responses for each of the subscales of the questionnaire. Overall, subjects ranked the redesigned version between neutral and agree on system usefulness (sysuse), information quality (infoqual), and interface quality (interqual). Conversely, the subjects ranked the original version in all categories between disagree and neutral. Overall the subjects were more pleased with the redesigned version than the original version.

4.10. Step 10: Make final modifications for program release

The final step in the process of redesign after completing all studies was making the final modifications based on the results of the studies. This iterative process as shown in the redesign cycle in Fig. 4 allows changes to be made in the redesign process before rollout. The shaded sections are those stages that need to be accomplished in the design of any system. The un-shaded areas represent the added methods in the process of redesign. Changes after rollout are very expensive to make.

5. Discussion

The methodology we employed in this redesign process showed that applications designed without regard to human-centered design guidelines can be successfully redesigned, although with significant costs. The methods we used herein can have important benefits toward user acceptance and thus, use of the program. The end results showed a significant difference in user performance and satisfaction between the original and redesigned versions of this program.

The steps from the analysis of the original version of the application through the comparison of the original and the redesigned versions was time consuming and would not have been needed if the other analyses had occurred during the design of the original application. However, the analyses of the original application allowed us to determine the overall functionality and usability problems with the system and provided direct evidence that the system needed to be redesigned not only from a good design point of view, but from the input provided by the users as well. The comparison of the original with the redesigned application provided the evidence that the redesigned application was superior to the original application and was therefore a necessary step, to show the importance of the work put into the redesigned application. The results of the usability questionnaire revealed that the subjects found the redesigned version far better than the original version in terms of user satisfaction, efficiency, and productivity, ease of learning, information display, and system functionality.

Analyzing the potential users of the system was necessary to determine the needs of the overall user population

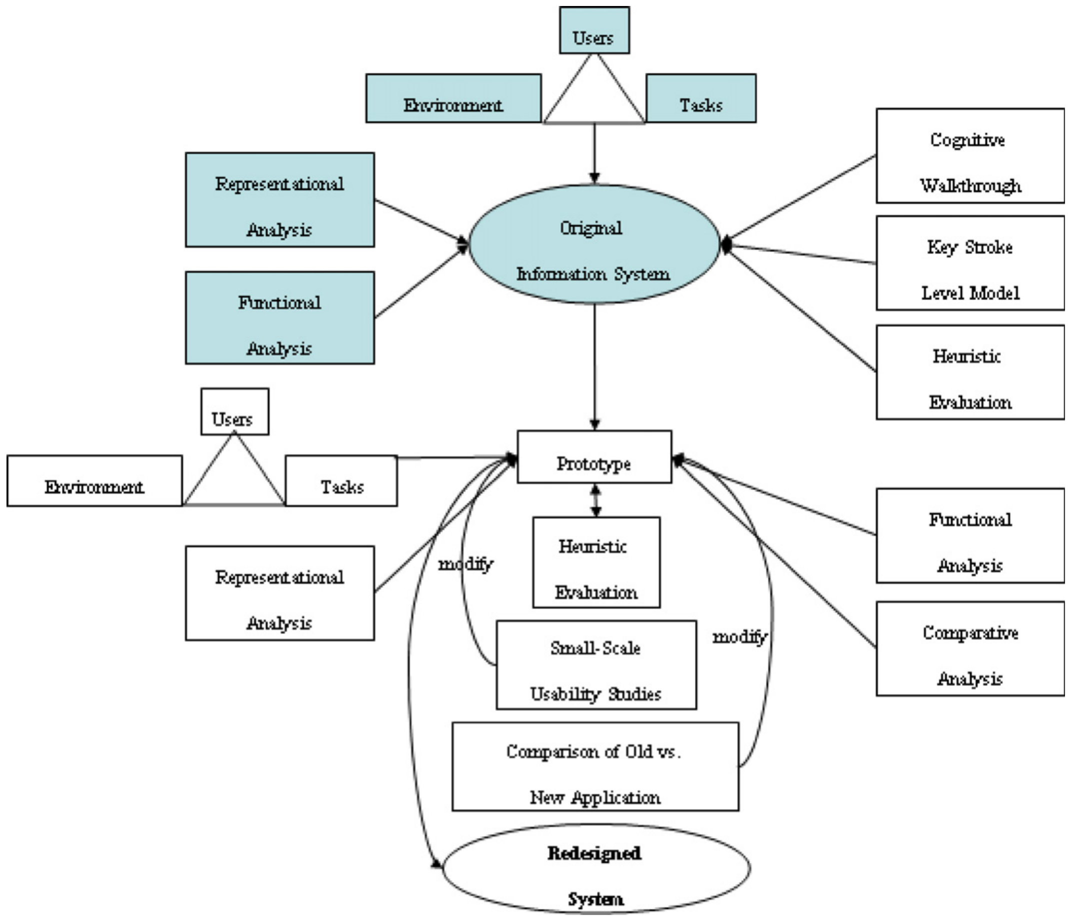

Fig. 4. Redesign lifecycle.

and not just the population using the program at a particular institution. This survey demonstrated that user analysis through direct-mail questionnaires is a proficient way to obtain information about potential users of a particular system, since the first step in the design of a successful product or redesign of an application is to understand the needs of the user population. This survey enabled us to determine important characteristics of the intended user population and directly affected the redesign of the family history tracking interfaces. For example, since the respondents reported that drawing a pedigree free hand is the most frequently used collection method, it was decided that a direct manipulation interface would probably best mimic this process. Since approximately one third of the respondents reported that their work environment is distractive, we designed an interface that supports easy resumption of tasks. Based on this survey, it was determined that nearly all respondents could potentially use specialized software to increase their productivity. Additionally, since there was a discrepancy in computer usage, the redesigned interfaces were designed to support novice, intermediate, and expert users.

The remaining steps which included the comparative analysis, the functional analysis, the representational analysis, creation of the prototype, small-scale usability studies, modification of the prototype and completion of the final application should be considered as usual steps in the design of all human-centered applications.

The redesigned system with its direct manipulation and form-fillin interface appears to bridge what Norman [45] terms the "Gulf of Execution" and the "Gulf of Evaluation." The "Gulf of Execution" was bridged by creating a direct manipulation interface that matched the goals of the user. The "Gulf of Evaluation" was bridged by allowing the user to draw the pedigree as they were creating a family on the screen thus interpreting and evaluating their results immediately. The more successful an interface is with bridging these gulfs, the more it decreases cognitive effort and increases the users' feeling of control. The real-world metaphor that we used in the redesign vs. the conversation metaphor of the original design allowed the users "direct engagement" [28] where the users sensed that they were manipulating the objects central to the task itself. We were able to determine from the responses of the users that we successfully minimized the "distance" between the goal of the user (drawing a pedigree) and the way that task was accomplished through the design of the interface [28]. The results of this study showed that we were successfully able to redesign a system that matched the real world tasks of the users and is easy to learn and easy to use. We believe the resulting redesigned system will allow the end-users to concentrate on the problem domain and not on the interface itself.

6. Conclusion

This paper demonstrates, through a case study, a framework for redesigning health care interfaces that incorporates well-documented user-centered design principles. Specifically, it demonstrates how a system that was designed without regard to user-centered design guidelines can be redesigned using this framework to create a system that models the characteristics and tasks of the users, thus increasing user satisfaction. The methods we employed in our framework show the benefits toward system usefulness, information quality, and interface quality. Health care is an intense information gathering activity and has a very complex organizational structure. Understanding the users, the environment, and their tasks is just the beginning when designing software or redesigning interfaces. Incorporating user-centered design principles throughout the design life cycle has the promise to assist in providing quality health care systems.

In addition to providing a framework for redesign process, three recommendations are proposed herein. Number one, there needs to be collaboration among administration, computer scientists, human factors engineers, cognitive scientists, and clinicians to ensure that health care applications remain intuitive and invisible, yet augment the tasks faced by clinicians. Number two; information technology groups need to be educated on the principles of user-centered design. Throughout this process of redesign, the message and solutions to user-centered problems from the technology group was "the users will just have to learn these steps." The answer should be, "No, the users should not have to learn these steps, they should be easy to understand." Interfaces need to be intuitive and promote exploratory learning and not provoke fear in the users of making irreversible errors. Third, the user culture needs to be educated not to tolerate poorly designed systems. How often it was heard from the users throughout the redesign cycle, "I am just so dumb when it comes to using computers." Users need to understand that the majority of the problems they encounter with poorly designed systems are not their problems, but are due to inattention to user-centered design guidelines. It is uncomplicated for a designer to create a system without consideration of whether the system is easy to navigate, easy to learn, easy to use, and easy to remember. Problems with user productivity, user satisfaction, user acceptance and utilization, user errors, user frustration, and user training requirements are often caused by mismatches between the designers' mental model and the users' mental model of the system. To achieve improvements in the quality of health care and reduce errors, researchers and system developers must work together to integrate the knowledge of user-centered design into the design of new systems. With modest effort, and attention to user-centered design guidelines, there is promise in providing quality health care applications, so that clinicians can focus on integrating the knowledge gained from the use of these systems and not on the mechanics of these systems.

Acknowledgments

This work was supported by the Texas Higher Education Coordinating Board under the Advanced Research Program, Grant No. 011618-0077-1999, and under work supported by the National Library of Medicine Applied Informatics Fellowship, Grant No.1F38 LM007188-01. We would also like to acknowledge the work that the Master's students, Rita Torkazeh, M.S., Yan Xing, M.D., M.S., and Lan Yang, M.S. contributed to this project.

References

[1] Smith MF. Software development: prototypically topical. Br J Healthcare Comput Informat Manag 1993;10(6):25–7.
[2] Sittig DF, Stead WW. Computer-based order entry: the state of the art. J Am Med Inform Assoc 1994;1(2):108–23.
[3] Williams D, Kennedy M. A framework for improving the requirements engineering process effectiveness. In: Proceedings of the 9th Annual International Symposium on Systems Engineering 1999. International Council on Systems Engineering: Brighton, England, June 6–11; 1999.
[4] Lederer AL, Prasad J. Nine management guidelines for better cost estimating. Commun ACM 1992;35(2):51–9.
[5] Pressman RS. Software engineering: a practitioner's approach. - New York: McGraw Hill; 1992.
[6] Mayhew DJ. The usability engineering lifecycle: a practitioner's handbook for user interface design. San Francisco: Morgan Kaufmann; 1999.
[7] Bates DW, Kuperman GJ, Rittenberg E, Teich JM, Fiskio J, Ma'luf N, et al. A randomized trial of a computer-based intervention to reduce utilization of redundant laboratory tests. Am J Med 1999;106(2):144–50.
[8] Tang PC, Patel V. Major issues in user interface design for health professional workstations: summary and recommendations. Int J Biomed Comput 1994;34:139–48.
[9] Tierney WM, McDonald CJ, Martin DK, Rogers MP. Computerized display of past test results effect on outpatient testing. Ann Intern Med 1987;107(4):569–74.
[10] Kreitzberg C. Managing for usability. In: Alber, Antone F, editors. Mulitmedia: a management perspective. Belmont, CA: Wadsworth; 1996.
[11] Shneiderman B. Designing the user interface. Strategies for effective human–computer interaction. Reading, MA: Addison Wesley Longman; 1998.

[12] Nielsen J. Usability engineering. Boston: Academic Press; 1993.
[13] Welch DL. Human factors usability test and evaluation. Biomed Instrum Technol 1998;32(2):183–7.
[14] Strasberg HR, Tudiver F, Holbrook AM, Geiger G, Keshavjee KK, Troyan S. Moving towards an electronic patient record: a survey to assess the needs of community family physicians. In: Proceedings—AMIA Symposium; 1998. p. 965–9.
[15] Hackos JT, Redish JC. User and task analysis for interface design. New York: Wiley; 1998.
[16] Bouwhuis DG. Parts of life:configuring equipment to individual lifestyle. Ergonomics 2000;43(7):908–19.
[17] Kirwan B, Ainsworth LK, editors. A guide to task analysis. London: Taylor & Francis; 1993.
[18] Vicente KJ. Cognitive work analysis. toward safe, productive, and healthy computer-based work. Mahwah, NJ: Lawrence Erlbaum Associates; 1999.
[19] Redish JC, Wixon D. Task analysis. In: Jacko JA, Sears A, editors. The human–computer interaction handbook Fundamentals, evolving technologies and emerging applications. Mahwah, NJ: Lawrence Erlbaum Associates; 2003.
[20] Denzin NK, Lincoln YS. Handbook of qualitative research. Thousand Oaks, CA: Sage; 2000.
[21] Miles MB, Huberman AM. Qualitative data analysis. Thousand Oaks, CA: Sage; 1994.
[22] Kieras D. Task analysis and design of functionality. In: Tucker AB, editor. The computer science and engineering handbook. Boca Raton: CRC Press; 1996.
[23] Welch DL. Human factors analysis and design support in medical device development. Biomed Instrum Technol 1998;32(1):77–82.
[24] Zhang J, Norman DA. Representations in distributed cognitive tasks. Cog Sci 1994;18(1):87–122.
[25] Woods DD. The cognitive engineering of problem representations. In: Weir GRS, Alty JL, editors. Human–computer interaction and complex systems. London: Academic; 1991. p. 169–88.
[26] Rasmussen J, Vincente K. Coping with human errors through system design: implications for ecological interface design. Int J Man Mach Stud 1989;31:517–34.
[27] Wickens CD, Andre AD. Proximity compatibility and information display: effects of color, space, and objectness on information integration. Hum Factors 1990;32:61–78.
[28] Hutchins EL, Hollan JD, Norman DA. Direct manipulation interfaces. In: Norman DA, Draper SW, editors. User centered system design: new perspectives on human computer interaction. Hillsdale, NJ: Lawrence Erlbaum; 1986. p. 87–124.
[29] Zhang J, Patel V, Johnson KA, Smith JW. Designing human-centered distributed information systems. IEEE Intell Syst 2002;September/October:42–7.
[30] (a) Kushniruk AW, Patel VL. Cognitive and usability engineering methods for the evaluation of clinical information systems. J Bio Inform 2004;37:56–76;
(b) Nielsen J, Mack RL, editors. Usability inspection methods. New York: Wiley; 1994.
[31] Cockton G, Lavery D, Woolrych A. Inspection-based evaluations. Fundamentals, evolving technologies and emerging applications. In: Jacko JA, Sears A, editors. The human–computer interaction handbook. Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers; 2003.
[32] Zhang J, Johnson TR, Patel VL, Paige DL, Kubose T. Using usability heuristics to evaluate patient safety of medical devices. J Biomed Inform 2003;36:23–30.
[33] Molich R, Nielsen J. Improving a human–computer dialogue. Commun ACM 1990;33(3):338–48.
[34] Jeffries R, Miller J, Wharton C, Uyeda K, editors. User interface evaluation in the real world: A comparison of four techniques. New York: ACM Press; 1991.
[35] Dumas JS, Redish JC. A practical guide to usability testing. Portland, OR: Intellect Books; 1999.
[36] Card SK, Moran TP, Newell A. The psychology of human–computer interaction. Hillsdale, NJ: Lawrence Erlbaum Associates; 1983 [Kieras D. Using the keystroke-level model to estimate execution times. Unpublished report. Ann Arbor: Department of Psychology, University of Michigan; 2001].
[37] Wharton C, Rieman J, Lewis C, Polson P. The cognitive walkthrough method: a practitioner's guide. In: Nielsen J, editor. Usability inspection methods. New York: Wiley; 1994. p. 105–40.
[38] Kushniruk AW, Patel VL. Cognitive computer-based video analysis. In: Greens R, editor. Proceedings of the Eighth World Conference on Medical Informatics; 1995. p. 1566–69.
[39] Lewis JR. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. Int J Hum Comput Interact 1995;7(1):57–78.
[40] Kirakowski J, Corbett M. The software usability measurement inventory. Br J Educ Technol 1993;24(3):210–2.
[41] Chin JP, Diehl VA, Norman KL. Development of an instrument measuring user satisfaction of the human–computer interface. In: Proceedings CHI '88-Human Factors in Computing Systems, ACM. New York; 1988. p. 213–8.
[42] Johnson C, Amos C. Kinsys: a database management application. In: Proceedings AMIA 99 Annual Symposium. Washington, DC, November 6–11; 1999.
[43] Johnson C, Johnson T, Zhang J. Increasing productivity and reducing errors through usability analysis: a case study and recommendations. In: Proceedings AMIA Symposium; 2000. p. 394–8.
[44] Perlman G. Web-based user interface evaluation with questionnaires; 1998. Available from: http://www.acm.org/~perlman/question.html.
[45] Norman DA. The design of everyday things. New York: Doubleday; 1988.