Lessons from Real-Life Usability Engineering in Hospital: From Software Usability to Total Workplace Usability

[1]Andreas Holzinger and [2]Hubert Leitner
[1]Institute for Medical Informatics, Statistics & Documentation (IMI)
Medical University Graz (MUG)
[2]Unternehmens Informations Management (UIM)
Steiermärkische Krankenanstalten Gesellschaft (KAGes)
andreas.holzinger@meduni-graz.at, hubert.leitner@kages.at

Abstract:
*In this paper the authors report about their experiences gained and the lessons learned during real-life usability engineering efforts within selected applications of openMEDOCS, which is a large Hospital Information System (HIS) in use at Graz University Hospital, one of the largest in Europe. OpenMEDOCS is a functional combination of several SAP based products including IS-H (patient administration), IS-H*Med (medical documentation), a SER-Archive and several KAGes specific individual developments. The HIS will provide services for about 10,000 end-users and links 20 hospitals. The system can be customized to individual requirements within the Hospital. Since the system is already in operation, a full User Centered Design from scratch is no longer possible. Consequently, the idea was to redesign the front-end functionalities by the application of workflow analysis and thinking aloud studies, including the end-users into the process in a real-life setting. As a result of these studies, and what we are able to learn from them, the system can be improved to provide the best possible solution to fit the end-users needs and requirements. During the studies, we recognized that the whole workflow must be considered. During such a holistic approach we found not only solutions within the software, but also, various usability problems in the environments. We called this approach Total Workplace Usability (TWU).*

Keywords: Usability Engineering, Existing Software Systems, Hospital Information System (HIS), ACM Classification: D Software Engineering; D.2.5.t Usability testing; J.3 Life and Medical Sciences; J.3.c Medical information systems

1. Introduction

MEDOCS stands for MEdical and nursing DOCumentation network of Styria and is the project name for the implementation of a Hospital Information System (HIS), started in the fall/autumn of 1999 [4].

The software products used are: IS-H from the company SAP, IS-H*Med from the companies T-Systems Austria and GSD; and, as communication servers, the SER Archive and E-Gate.

The project was started 1999 with a pilot implementation. For the implementation in the 20 KAGes hospitals, the execution was divided into to three levels:

- The *central (core) area* is the obligatory section installed in all hospitals. It contains the patient administration, accountancy and Diagnostics Findings Report (DFRs) creation;
- The *basis software* covers the individual products, which can be installed to provide functionality under MEDOCS products, e.g. the OP Module and the Healthcare Module.
- The *special range* contains all functionalities, which can be provided by sub-systems or which can be used, for example, for creating parameterized documents using IS-H*Med tools.

The decision, what and how much to install, must be made by each hospital individually. At present, the rollout of the core functionality and the basis software has been finalized.

A further aim is to optimize the user interface of all parameterized documents in the actual software design in order to safe time and expensive iterations at a later date.

2. Usability in Hospital Information Systems

In a recent article Jakob Nielsen described usability problems of Hospital Information [10]. In his article he described – often with reference to [8] – some interesting issues:

Misleading Default Values. According to Nielsen, years of usability studies in many domains have shown that users tend to assume that the given default or example values are applicable to their own situations.

New Commands Not Checked Against Previous Ones. According to Nielsen, if users are doing something which they have already done, the system should ask whether both operations should remain in effect or whether the new command should overrule the old one.

Poor Readability. According to Nielsen, because patient names appear often in small fonts, which are difficult to read, it is possible to select the wrong patient. However, it is a fact that too much information on the screen is difficult to read.

Memory Overload. The commonly known limits on human short-term memory make it impossible to remember everything given an abundance of information. According to Nielsen, humans are poor at remembering exact information, and minimizing the users' memory load has long been one of the top-ten usability heuristics. Facts should be restated when and where they're needed, rather than requiring users to remember things from one screen to the next.

Overly Complicated Workflow. According to Nielsen, many aspects of the system required the end-users to go through numerous screens that conflicted with typical hospital workflows. As a result, the system wasn't always used as intended. Nurses, for example, often keep separate paper records. Generally, wherever users resort to sticky notes or other paper-based workarounds, it is obvious that the Information System fails to support the workflow properly.

It is a matter of fact that Usability in the Hospital is of vital importance and is a growing area where a lot of research will be necessary in the future. But – and this is a large but – we are of the opinion that only looking at the software system is *not enough.* The total workflow must be considered.

To gain insight into the behavior of end-users, when performing tasks within real-life settings, was the aim of our experiments.

3. Experimental Setting

In the first run eight experiments took place within *real-life settings* (see figure 2) of various medical departments within Graz University Hospital and General Hospital Graz-West. For each of the tasks one experienced medical doctor (we call them: expert end-user) and one rather inexperienced medical doctor (we call them: novice end-user) were chosen, aiming at carefully finding out whether differences in time-to-perform tasks resulted from the end-user's expertise, from the graphical user interface (GUI) or from other environmental circumstances.

Methods

Out of the palette of various usability engineering methods [6], we used the *thinking aloud method.* According to Nielsen [11], thinking aloud is one of the most valuable usability engineering methods. Basically, a thinking-aloud test involves having an end-user use the system whilst continuously thinking out loud [12] [9] [2]. The roots of this method go back to the first studies of human problem solving behavior performed in 1945 [3].

By *verbalizing their thoughts,* the test users are able to understand how they view the computer system and this makes it easy to identify the users' major misconceptions. Designers gain a very direct understanding of which parts of the dialogue cause the most problems because the thinking-aloud method shows how users *interpret* each individual interface item [14] [15] [1], [5].

A variant of the thinking aloud method, which we found to be very helpful in hospital settings due to the fact that many tasks are done in cooperation with others, is called *constructive interaction* and involves having two test users use one system together.
The theoretical basis for this method is co-discovery learning [7] [13]. During a co-discovery learning session, two or more participants attempt to perform the tasks together whilst their interaction is observed. As the participants complete tasks, the tester encourages them to explain what they are thinking. The main advantage is that the test situation is much more natural than standard thinking aloud tests with single users working alone, since people are used to verbalizing their thoughts when trying to solve a problem together. Therefore, users may make more comments when engaged in constructive interaction than when simply thinking aloud for the benefit of an experimenter.

Equipment

Software

The software examined was part of the customized medical documentation system IS-H*Med, which was developed as a parameterized document. This module is used for the documentation of those parameters from the intensive care area, which are legally prescribed for the clinical efficiency evaluation of the intensive care area, to be collected and forwarded in a digital form.

The module contains an abundance of different input fields. The documentation takes place almost exclusively in a structured form. The data are partially transferred from the PDM system by hand. Diagnoses and clinical achievement report are transferred using the software DIACOS, which is transmitted via an IS-H*Med interface.

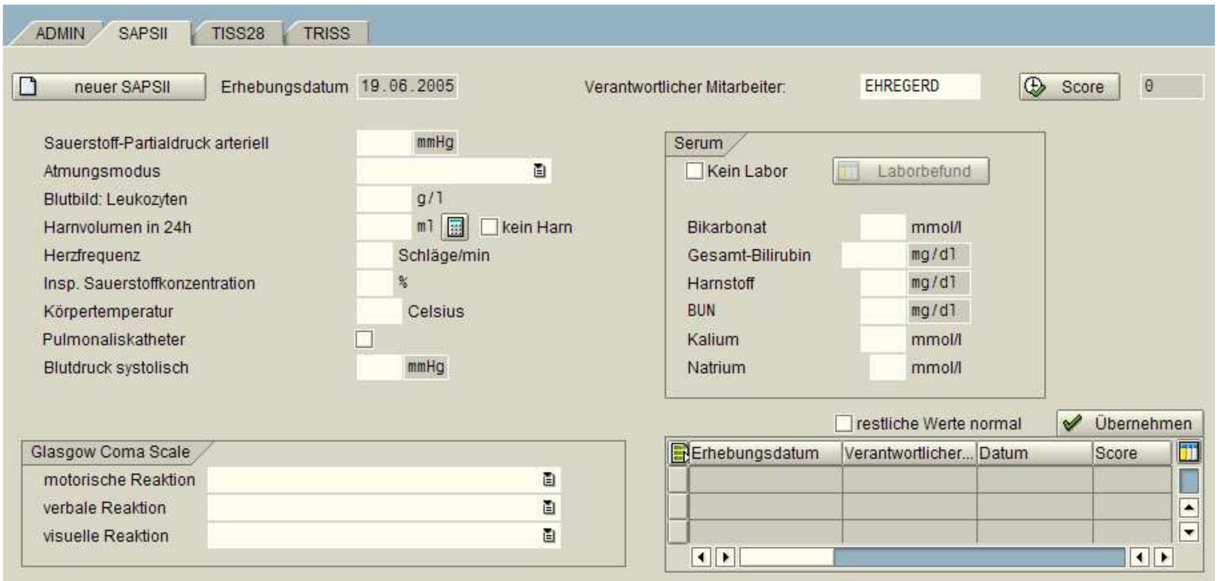

Figure 13. Typical Screenshot of the System

Hardware

The hardware at the documentation work areas usually consists of two standard personal computers. The PDM system runs on one and the intensive care module and the DIACOS on the other.

Video equipment

Two digital Sony HC 40 Video cameras, a button-hole microphone and a solid stands, and one mirror were used. The synchronization of the two videocameras was done manually (with a clapboard).

Tasks

Typical daily tasks were researched. At first, the time to complete the task was estimated by the expert. Example: "*Die Aufgabe besteht darin, für einen Patienten der Intensivstation die täglich notwendigen Daten im Intensivmodul zu erfassen*" (The exercise consists of entering the daily necessary data for a patient in the intensive care module), which was estimated at approximately 10 to 12 minutes by the expert and actually took 17 minutes. During this experiment there were 3 interruptions due to telephone calls, with a length of about 2 minutes each. Taking these disturbances into consideration, the time to complete the task was almost exactly as estimated. The expert was extremely concentrated and in complete control of his actions. He carefully optimized his operation steps in a systematic manner and in an optimal sequence. Due to his end-user behavior, we could claim that he was a *best practice test-user*.

Every expert end-user was followed by a *novice end-user,* who carried out the task under exactly the same conditions. Example: The time to complete the task was 31 minutes. As later seen in the evaluation session the test user seemed to be nervous, which was partially due to cognitive overload, caused by the abundance of information. He seemed to be far less communicative in the thinking-aloud and was cautious in making his medical decisions. Most of all, it was notable that this test person had difficulties with a badly adapted working place, insufficient room for the mouse, etc.

4. Lessons Learned: Total Workplace Usability

In the analysis phase, everyone who took part in the experiments (software engineers, usability engineers etc.) carefully watched and discussed the videos taken. Some very interesting facts could be derived by analyzing the videos, which astonishes especially the software engineers: it came through – consistent in all experiments – that the main problems were *not* in the HIS system itself but rather in the whole workflow. Derived from our experiences we can summarize that besides the Software Usability aspects usually considered, a Total Workplace Usability must include:

- *End-user Training* to familiarize them with the software being used; Goal: reduce time to perform task to (in one of our examples to under 10 minutes) – which seems to be possible. Benefit (in our specific example): Three tasks could be undertaken in the same time, total cost saving for the Hospital;
- *Electronic Tutoring* to assist part-time end-users to find their way through the workflow quickly and to help them to solve their problems rapidly. Benefit: once developed, it would constantly assist *many* end-users at all workplaces, independent of end-users training efforts;
- *Customizing and proper adaptation* of third-party Software (DIACOS, CareVue), Benefit: Faster information gathering by the end-users would increase work speed and reduce dissatisfaction due to cognitive overload.
- *Ergonomic Aspects* of the workplace include: proper distance to the screen, correct table height, proper chair (easily adjustable for height, to enable the user to quickly achieve a comfortable angle from eyes to the screen), proper mouse pad location and working space. Benefit: End-users can concentrate and feel more comfortable during the strenuous information processing process; results are immediate in lower task performance time and consequently a better output.

Figure 2. An example of a typical setting during our usability studies.

5. Conclusion

Our experiments with end-users in real-life settings in the Hospital showed that only minor adaptations of the GUI of the developed software were necessary. However, all experiments showed that the investigated software was not the primary cause of long task performance times. It was rather the whole workflow, which included the work with third-party products, work in a disturbing environment and interestingly, human factors of the workplace, including hardware usability, seating and mouse space. To ensure a *total acceptance* of a system, satisfaction and most of all the *reduction in time to perform task* is essential. To reach that, all issues must be considered which we call Total Workplace Usability (TWU). This also must include end-user training, electronic tutoring, customizing and adaptation and integration of third-party software and ergonomic aspects of the workplace. Consequently, it is essential to include software usability, psychological, behavioral-cognitive, ergonomic, team oriented, organizational and environmental issues into the focus of research in close cooperation with the software developers. This brings immediate benefits for medical personnel, the hospital and finally for the patients. However, a lot of work must be carried out to raise awareness for the importance of usability engineering efforts.

6. Acknowledgements

Thanks to all medical doctors who participated in our usability studies and allowed an exciting insight into their daily workflows. Thanks to all engineers from department I5, who took part in these studies, helped to analyze the material, for valuable discussions and their reengineering efforts. Finally, thanks to Markus Pedevilla, who promoted these usability studies within the Hospital.

7. Glossary

openMEDocs … refers to the functional combination of the products IS-H, IS H*Med and SER archives and also incorporates KAGes specific developments:

- IS-H = administrative basis (in particular the patient admission, transfer, dismissal and accounting);
- IS H*Med = medical documentation;
- SER archives = combined digital archives.

DIACOS … is the product name of a system for the support of diagnoses, medical documentation and clinical achievement reports.

CareVue … is the product name of a specific PDM System.

PDM System … Patient Data Management System; usually Software for recording vital parameters of patients in intensive care.

8. References

1. Andrews, K.: Web Usability on the Cheap. in Holzinger, A. ed. Human-Computer Interaction in the 21st Century, Austrian Computer Society, Vienna, 2001, 83-95.
2. Bereiter, C. and Bird, M.: Use of thinking aloud in identifcation and teaching of reading comprehension strategies. Cognition and Instruction, 2. 131-156.
3. Duncker, K.: On problem-solving. in Dashiell, J.F. ed. Psychological Monographs of the American Psychologoical Association, Vol. 58, APA, Washington (DC), 1945, 1-114.
4. Gell, G., Madjaric, M., Leodolter, W., Kole, W. and Leitner, H.: HIS purchase projects in public hospitals of Styria, Austria. International Journal of Medical Informatics, 58-59. 147-155.
5. Holzinger, A.: Experiences with User Centered Development (UCD) for the Front End of the Virtual Medical Campus Graz. in Jacko, J.A. and Stephanidis, C. eds. Human-Computer Interaction, Theory and Practice, Lawrence Erlbaum, Mahwah (NJ), 2003, 123-127.
6. Holzinger, A.: Usability Engineering for Software Developers. Communications of the ACM, 48 (1). 71-74.
7. Kennedy, S.: Using video in the BNR usability lab. SIGCHI Bulletin, 21 (2). 92-95.

8. Koppel, R., J.P., M., Cohen, A., Abaluck, B., Localio, A.R., Kimmel, S.E. and B.L., S.: Role of Computerized Physician Order Entry Systems in Facilitating Medication Errors. Journal of the American Medical Association, 293 (10). 1197-1203.

9. Lewis, C. and Mack, R.: Learning to use a text processing system: Evidence from thinking aloud protocols. In SIGCHI conference on Human factors in computing systems, (Gaithersburg (MD), 1982), 387-392.

10. Nielsen, J.: Medical Usability: How to Kill Patients Through Bad Design In: Jakob Nielsen's Alertbox, April 11, 2005, 2005.

11. Nielsen, J.: Usability Engineering. Morgan Kaufmann, San Francisco, 1994.

12. Nisbett, R.E. and Wilson, T.D.: Telling More Than We Can Know: Verbal Reports on Mental Processes. Psychological Review, 84 (3). 231-259.

13. Rubin, J.: Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. Wiley, New York, 1994.

14. Spool, J.M., Snyder, C. and Robinson, M.: Smarter usability testing: practical techniques for developing products. In Conference companion on Human factors in computing systems: common ground, (Vancouver, British Columbia, Canada, 1996), 365-366.

15. Waes, L.V.: Thinking Aloud as a Method for Testing the Usability of Websites: The Influence of Task Variation on the Evaluation of Hypertext. IEEE Transaction on Professional Communication, 43 (4). 279-291.