INITIATING USABILITY METHODS WITH A NEW ENGINEERING DESIGN TOOL

AITA SALASOO

Abstract:

Goals, methods, and results of initial usability work for a new, intelligent software product are described. The approach yielded a number of expected and unforeseen benefits, as well as lessons for

INTRODUCTION

This work was done on a new software product being developed by Bellcore for use by Bell Operating Company engineers. It is an intelligent tool that supports network design problems by integrating knowledge from several corporate (locally controlled) databases and using models containing that knowledge to guide the design process. An example design problem for a complex piece of network electronics is determining the list of components required and instructions for their assembly. The product allows information and expertise concerning new technology, standards, vendors, inventory, engineering rules, policy, and geography to be used to solve a variety of engineering design problems without overloading the memory and knowledge demands of an individual user.

We have begun to implement a usability approach to user interface requirements and design on the first release of the product. Our approach is based on the Whiteside et al. (1988) model, beginning with usability objectives, followed by low-fidelity prototyping and iterations of testing and redesign. This report describes work up to the first iteration of usability testing. One human factors engineer, a Macintosh® computer with HyperCard®, and a videocamera were the core of this usability effort.

APPROACH TO USABILITY

User interface requirements had two components. First, a section of a multipurpose requirements document coauthored by the whole project team documented user functionality requirements, the user interface style and components, usability objectives, and an initial usability test plan. The document helped to achieve understanding of the usability approach planned for the entire project cycle by the project team and the client managers.

In addition, the other user interface requirements deliverable was a tested usability prototype. The prototype was an excellent communication vehicle with developers, users, and a marketing tool. The usability test report that accompanied the prototype served as a model for further work on the product, as well as for other projects implementing usability methods. The benefits of this approach to user interface requirements are schematized in Figure 1.

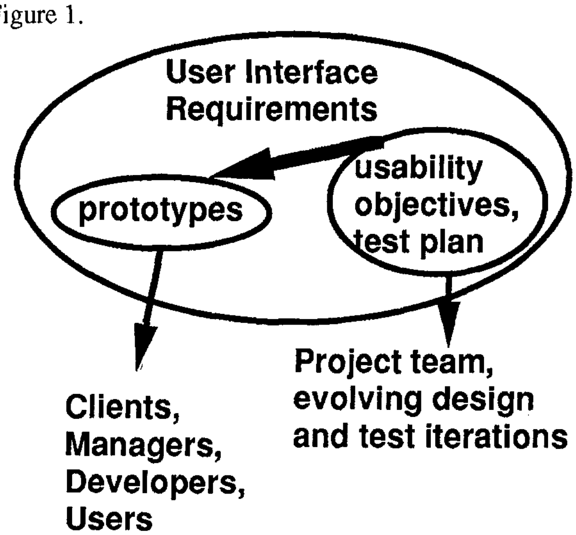

Figure 1. Approach to User Interface Requirements

Usability Objectives

Thirty attributes and subtasks were originally specified in accordance with the Whiteside et al. tabular approach. Given the new product domain, some specification tables were incomplete where baseline measures were missing. However, two complete design task scenarios were specified to test 17 of the attributes in the first iteration of testing. Both performance and subjective data were gathered where appropriate. Our view of usability objectives was as an evolution of attributes and measures.

First Usability Prototype

The usability prototype was the first complete vision of the complex and abstractly-conceived engineering tool. As dictated by generic artificial intelligence software architectures, it separated the user interface from domain knowledge, both in user guidance methods and in display generation. Context-sensitive help and online tutorials for the user interface and for design engineering domains were separated. User interface displays used menus where user inputs were known by the model, fields where they were open-ended (e.g., names or nonrestricted numbers), software buttons for optional or forced-choice inputs, and selectable graphical displays to present spatial information. This simplification allowed memory loads and error opportunities to be minimized for user inputs, while being generic enough to generalize across domain models.

The prototype allowed users to complete two realistic design engineering jobs successfully after 20 minutes of experience with two online tutorials. Seven engineers participated in the initial usability testing with this prototype, each for a half-day session. We collected performance and subjective data on initial acceptance, errors, guidance needs, and time to complete tasks using keystroke, video, questionnaire, and interview techniques.

RESULTS

Initial testing added baseline values missing from the original usability objectives specification tables. In addition, testing identified major user training and acceptance issues, as well as strengths and weaknesses of the initial design. More than one-third of the original problem time was decreased by design changes made after four subjects. The subjective data added significantly to the task performance measurements. The first iteration resolved broad issues and identified areas requiring attention in redesigns and further testing.

Design Changes and Issues

Initial testing results supported the proposed task flow, the menu and button based user interface style, the use of graphics, and other aspects. For example, the flow of the class of design problems tested was reflected in the menu structure of the application and was taught explicitly in the domain tutorial. It was tested in three ways: Navigation errors between the broad parts of the application made by usability testing participants were counted. A post-performance test asked subjects to map out the basic sequence of completing a design job with the tool. Finally, a questionnaire item asked participants to rate how straightforwardly tasks could be performed. All measures supported the flow, and no participants made negative comments about it during the interview.

Among the issues demanding more attention after this testing were user inputs with both mouse and keyboard, the contents and access of tutorials and help, report contents and format, and terminology. While mouse use for selection was easy and preferred by participants, some difficulty was experienced when more than three keyboard inputs were required on a screen. The user group has little experience with mice today, and the presence of two cursors was confusing. This pointed to the need for additional measures to overcome these mouse-beginner problems. Omissions from the tutorials, e.g., how to print screens, were identified by early testing participants and fixed. Similarly, incomplete help access methods were uncovered during early testing.

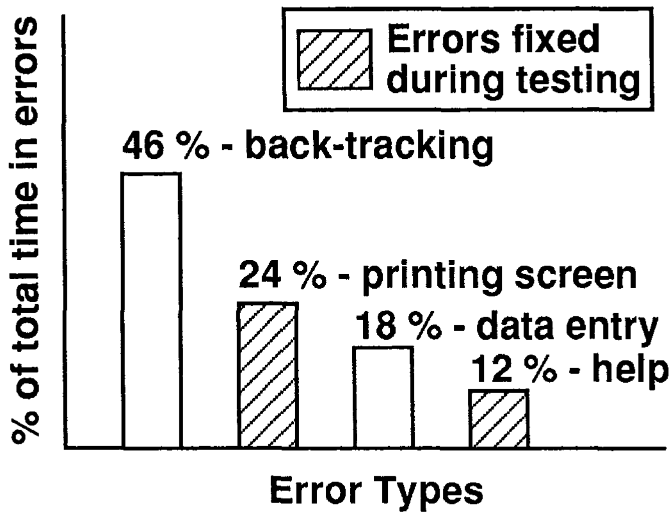

Figure 2. Time Spent in Errors

Figure 2 shows some results of our first usability testing, in terms of time spent in errors during task performance. The printing and help errors, as described, were fixed during testing. The data entry errors relate to the mouse issues above, as well as to display complexity issues discussed in the next section. Backtracking (within the current subtask of the application) is not strictly an error, but was documented as an exploratory behavior of new users. We intend to monitor it with more experienced users and anticipate its role to decrease.

Issues of Artificial Intelligence

The usability prototype architecture and testing results demonstrated several issues pertinent to intelligent systems and their user interfaces. Acceptance of a guiding, intelligent tool was not automatic in our user group, and testing results helped client managers appreciate the need for major steps to enhance initial acceptance of the tool. The dependence of the user interface on the underlying domain knowledge (and its structure) was demonstrated in the usability prototype. First, display organization was a direct consequence of the knowledge groupings in the models containing various bodies of engineering information that drive the application. These models in our case are created by others -- local domain experts who may not be knowledge engineers or user interface designers. The boundaries of acceptable display amounts and layouts were demonstrated by the usability testing results, i.e., the behavior of engineers trying to solve real problems with the prototype. In particular, screens requiring too much input led to data entry problems. This evidence brought home the dependence of the user interface on the domain models more effectively than previously tried methods. Similarly, during testing, users relied heavily on domain help. This part of user guidance is contained in the engineering expert models, again alerting client managers to special concerns relating to model building.

LESSONS

We plan to pursue systematic user-based performance and subjective assessment methods throughout the course of the product's project cycle. Our initial successes have led to the allocation of additional human and other resources to the usability approach to building complex software tools.

Four lessons that may help others applying usability methods to new software product development, whether or not for intelligent systems, follow:

First, minimizing the fidelity-gap (in terms of platform) between the first usability prototype and the final product will enhance the evolving cooperation with system developers. We learned this the hard way and hope others can bypass associated problems. Second, while collecting and analyzing several dependent measures for a given usability attribute may have associated human costs, the strength of converging evidence is worth it. We experienced the benefits of back-up data on conclusions that were not at first seen as important by developers. Also, in some cases, measures may yield contradictory results, pointing either to ceiling effects or to important discrepancies between preferences and performance. Third, we recommend communicating usability goals, methods, and results before showing the products or prototypes used as usability test tools. This helped focus our audiences on the user-centered nature of our efforts, rather than on their own speculations relative to the screen images they saw. Finally, explicit plans to retire usability prototypes from demonstrations are necessary to avoid over-reliance on early usability testing tools as marketing or other planning devices.

REFERENCE

Whiteside, J., Bennett, J., Holtzblatt, K. (1988) *Usability Engineering: Our Experience and Evolution.* In M. Helander (Ed.) **Handbook of Human Computer Interaction.** North-Holland: Elsevier Science Publishers.

ABOUT THE AUTHOR

Aita Salasoo

Aita Salasoo is a member of technical staff at Bellcore (Bell Communications Research, Inc.). She has a PhD in Cognitive Psychology from Indiana University and has published language and memory research, as well as applied studies of human factors in software development. At Bellcore, Aita has supported planning and engineering software systems as a human factors engineer, and has been active in the transfer of user-centered design and human factors technology throughout the company. Her current research addresses data access, visualization, and integration issues for intelligent network applications.