Special Session - User-Observation and Sketching as Core Stages in the Software Design Process

David Socha\textsuperscript{1} and Skip Walter\textsuperscript{2}

Abstract - This interactive session focuses on user-observation and sketching, two of the least practiced and taught components of good software design. Attendees will experience how a rapid cycle of user observation, ideation, presentation, and feedback can quickly produce excellent design concepts by taking advantage of multiple levels of group intelligence: individuals, small groups, and groups of groups. It is expected that participants will develop an understanding of why these steps are crucial to the software design process and an understanding of how to incorporate these steps into their software design courses.

Index Terms - Creativity, ideation, sketching, software design, user observation.

CONTEXT

As software becomes an ever larger percentage of the “Whole Product”\cite{1} in the Information Economy, a broader software design process becomes critical to the early and easy acceptance of a software application. Unfortunately, as a general rule, most software design models do not cover the entire range of activities required for good software design. Furthermore there are few courses that teach how to do software design.

In particular, most software design models omit two of the least practiced and taught components of good software design: user observation and sketching. The goal of this session is to have the attendees experience the value of these two important aspects of software design.

USER OBSERVATION

Many software products automate what users are doing, so it is vital to understand what they do before designing the product. Additionally, using software will require the users to change how they act. It is important to design the software to facilitate this transition. The organizational intervention nature of software design \cite{2}\cite{3}\cite{4} is one of the less well understood, yet important, aspects of software design.

A valuable technique for understanding the user’s current state is user observation. It is critical because people are very poor at observing themselves. If asked what they do, they will tell you something different from what they actually do. You can’t trust what they say. The only reliable way to know what users do is to observe them. User observation is an important step to do before the formal prototyping techniques that are more common in computer science.

IDEATION AND SKETCHING

Designing a desired product requires ideation: generating and exploring lots of new design concepts. The input to ideation is the understanding of the user’s current state, and a vision of their desired future state, including how you want the users to act differently in the future.

Sketching is a powerful technique for generating ideas in ideation. Sketching is the process of rapidly generating many possible designs, rather than being precise about any particular design. Sketches often are inconsistent, simple, and rough. Sketches are throw-away products created to enhance conversation within a small group of people.

Note that sketching is different from the formal diagramming that has received a great deal of attention in the computer science community. Formal diagrams are useful for understanding and reasoning about an existing system (e.g. for documentation, semantic analysis, debugging, or reverse engineering). They tend to be semantically precise. In many cases, the goal is for formal diagrams to be precise enough to be executed by computers (e.g. CAD). Formal diagrams often are “keepers” – artifacts that are valuable to keep and maintain for the project. Sketching fulfills a very different role and requires different techniques and different measures of success.

OVERVIEW

This session is devoted to practicing user observation and sketching, in order to generate a design for a product to help a specific set of users. The basic steps of this session are as follows.

Warm-up. We will divide the audience into small groups of 3-5 members to do a warm-up sketching exercise based on the popular game Pictionary. The goal of this exercise is to illustrate how quickly communication can occur with even very crude sketches.

Meet the users. Then we will provide guidance on how to do the observation, and then play a videotape of a representative user for a potential software solution.

Generating possible solutions. Following the playing of the videotape, each individual in each small group will use brainstorming techniques and concept sketching to generate 5-10 sketches. Each concept sketch will represent a possible...
solution to meet the needs of the user as pictured in the videotape. Following the generation of the concept sketches, the small groups will then generate a single sketch that captures the essence of the group’s individual sketches. This group concept sketch will then be presented to the large group. The small groups are encouraged to draw from the ideas presented from the other groups.

Revisiting the users. After this sharing of sketches, we will show an additional videotape of user research to further stimulate the ideation process.

Revising and generating to select a final solution. Working again in the small groups, each person will brainstorm and sketch an additional 5-10 potential prototypes. From these 15-50 sketches, the small group will select one to present to the larger group. After the presentations, each small group will compare their presented solution with the users’ needs (observed from the video) to determine how their prototype, if built, would change the user’s world. Each small group will then pick the most important difference, and present that to the whole group.

Reflection. To finish the session, each individual will spend a few minutes reflecting on the process of the session and draw a sketch of how sketching would improve the process of Software Design.

In this session, participants will:

- Do user observation from video ethnography.
- Do rapid sketching for creating designs, and for communicating designs.
- Experience the effectiveness of collaborating through an appropriate combination of individual work, small group work, and large group work.
- Experience the effectiveness of rapid cycles of generating and reducing using sketching, instead of list generation.

**ACKNOWLEDGMENT**

The authors thank Jeff McKenna, Steve Forgey, Barney Barnett, and Robin Adams for their thoughtful comments and insights into the software design process and design education which contributed to this article. This research was supported in part by NSF Grant number EIA-0121326, and by the Institute for Scholarship on Engineering Education through the NSF funded Center for the Advancement of Engineering Education (NSF ESI-0227558).

**REFERENCES**


**BIOGRAFICAL INFORMATION**

DAVID SOCHA, Ph.D.

David Socha studies the human side of software development. He currently is the Project Manager on the UrbanSim project, and a Lecturer at the Computer Science & Engineering department, both at the University of Washington, Seattle where he received his Ph.D. in 1991. After his doctorate, he spent 11 years in industry, 6 of those managing teams of software developers, before returning to practice software development in academia.

SKIP WALTER

Skip Walter is CTO of Attenex Corporation bringing over 35 years of technology product development experience along with executive management experience in Fortune 1000 companies and start-up businesses. Skip was the creator of DEC’s ALL-IN-1, a $1 billion revenue per year office automation system. He taught Masters and PhD courses in interactive product planning and tangible knowledge design at the Institute of Design at the Illinois Institute of Technology for 10 years.