



Product Usability: Survival Techniques

Jared M. Spool, Tara Scanlon, Carolyn Snyder

User Interface Engineering

800 Turnpike Street, Suite 101

North Andover, MA 01845

+1 508 975 4343

jspool@uie.com, tara@uie.com, csnyder@uie.com

ABSTRACT

Product developers are typically faced with small budgets, tight schedules, and over-committed resources. To deliver high-quality products under these constraints, developers need an understanding of basic design principles, techniques that allow them to work effectively with materials on hand, and a development process that is built around the use of such techniques. This workshop explains how low-fidelity prototyping and usability testing can be used in a process of iterative refinement in order to develop more usable products.

KEYWORDS

Design principles, usability testing, usability evaluation, prototyping, low-fidelity prototyping, process management, product development, practical techniques.

PART 1: DESIGNING FOR USABILITY

In product development and interface design, the notion of “getting it right the first time” is a myth. Rather than focus on getting something perfect from the beginning, we advocate a process of successive refinement that allows for frequent measurements against predefined goals.

The Cause of Usability Problems. Our research shows that the vast majority of usability problems come from a single cause: the development team didn't know a key piece of information. If they had known that information earlier, they would have designed the product to accommodate it, and the usability problem would never have occurred.

The Four Stages of a Prototype. In a sense, the current release of a product is just a prototype for the

next release, just as the previous release was a prototype for this release. Every prototype goes through the same four stages: *Plan*, *Implement*, *Measure*, and *Learn*. In the Plan stage, the team looks at their areas of highest risk for the product and determines what information they need for the key decisions they are facing [1]. During the Implement stage, the team builds just enough of the product so that they have something to test. In the Measure stage, the team collects data that will help them make decisions. Usability testing is one very effective way to collect this data. And in the Learn stage, the team responds to what they've observed by adjusting the interface, functionality, or schedule for the product.

Successful Iterative Refinement. The key to this four-stage process of successive refinement is to conduct the stages very quickly, ideally in a matter of hours rather than weeks or months. Implementation is often the most time-consuming stage, so reducing implementation time will allow a development team to go through more cycles, and discover key information earlier in the project.

PART II: PAPER MOCK-UPS

The purpose of a prototype is to provide feedback on a design. But implementing, testing, and shipping an entire release is an expensive way to get critical information. Using paper mock-ups substantially shortens the Implement phase, allowing more iterations than would be possible with electronic prototyping techniques.

The Basics of Paper Mock-ups. Using common office supplies, (markers, index cards, transparency film, etc.) the development team can quickly construct a fully-functional prototype of the product interface. This technique has proven effective even for sophisticated, high-tech products. We have found that, with a little imagination, you can effectively prototype almost anything!

Usability Testing with Paper Mock-ups. Usability testing of a paper mock-up is quite straightforward. Using their finger as a "pointing device," users can select from menus, click on buttons, and otherwise interact with interface elements. One or two members of the development team simulate the behavior of the computer, taking the appropriate action in response to the users' requests. Paper mock-ups are easy to modify, so the team can even make changes in the middle of a usability test!

Advantages of Paper Mock-ups. An iterative development process that employs low-fidelity prototyping is an excellent method for ending "opinion wars" and other project-killers. Instead of wasting time on subjective arguments, the development team can instead focus on objective usability goals (e.g., an untrained user can complete the installation within 5 minutes) and test different approaches until the first satisfactory one is found. At that point, the team can turn their attention to the next issue on the priority list.

Comparison with Electronic Prototypes. An electronic prototype (such as one developed in Visual Basic) seems more attractive at first glance because it has a more polished "look." However, when it comes to prototyping the "feel" of the product, i.e., its behavior, some form of programming is generally required. In a paper mock-up, the "feel" is simulated by the person playing the computer, so this programming time is eliminated. In our experience, most of the risky issues in product development pertain to the "feel" (Does the product have the right set of functions? Can users figure out how to do their work?) rather than the "look" (Are the icons clear? Does it conform to the style guide?).

Drawbacks of Paper Mock-ups. Paper mock-ups are not a cure-all. For example, it is not possible to assess response times with a paper mock-up, and it may be harder to get accurate feedback on the visuals of the product. Also, sometimes it is necessary to fake data that would be available in an actual running system. We generally recommend using paper mock-ups early in the product development process, when their advantages are greatest.

PART III: BASIC DESIGN PRINCIPLES

Product developers need a working knowledge of the key factors in good interface design. We have found

that a majority of usability problems can be addressed (or prevented) by careful attention to three principles: affordances, mental models, and tool time.

Affordances. Affordances [2] are the built-in "clues" that communicate to users how the interface is to be used. Buttons, icons, words, and other controls make up the visual language which helps users know what to do. In many cases, usability problems can be attributed to false or missing affordances.

Mental Models. Users will subconsciously develop their own cause-and-effect mental model of how an interface works [2]. By focusing on the users' mental model, designers will better understand which details of the product's inner workings should be communicated to users (or hidden from them!).

Tool Time. Tool Time is the portion of time that users spend being distracted by the interface ("how do I indent this paragraph?") instead of focusing on their own work ("what do I want this memo to say?"). Reducing tool time allows users to be more productive. Whenever developers can identify tool time, they can often minimize it through means such as eliminating steps or moving key functions closer together.

DESIGN COMPETITION

The design exercise and competition is the high point of the tutorial. Using supplied materials, participants work in small teams to construct a full working paper prototype of a fast food ordering kiosk. The designs are then evaluated by a series of usability tests, and each team has the opportunity to make changes after each round. The team that has the most usable interface wins fabulous prizes!!

The competition gives participants "hands-on" experience in the concepts presented in the course. Over 2,000 professionals have participated in past competitions, and many tell us that this is the best part of the course.

REFERENCES

1. Snyder, C. Managing Risk by Using Paper Prototypes in *Software Developer and Publisher*, Sep/Oct 1996, 38-40.
2. Norman, D. *The Design of Everyday Things*, Basic Books, New York, 1988.