Randy Almand
Id# XXXXXXX

Evaluating User Interface Designs

Virtually every application written to interact with a human has some form of a user interface. One of the most overlooked aspects of contemporary software design is proper user interface design and evaluation. Any developer of a non-trivial application will agree that testing software for correctness and adherence to specification is vital to success, although testing the system's interface may take a lesser priority to testing its primary functions. As software systems grow increasingly complex, testing every condition in a large system approaches impossibility.

Developers rely on guidelines constructed in the planning stages to verify if the system behaves as it should in any given scenario. Among these guidelines should be provisions for how the software should look and interact with a user. Testing that the system's user interface is consistent and behaves appropriately during normal usage as well as emergency situations is critical to the software's reliability and ease of use. If a software system meets its functional objectives but a typical user has difficulty carrying out a certain procedure, there needs to be some way of evaluating why this situation exists and how to correct it.

Why evaluate user interface designs?

By incorporating user interface design evaluations into software design projects, companies have gained insightful feedback, seen fewer emergencies as projects are preparing to ship, and finished many projects sooner while spending less money. (Shneiderman, 144) If those facts alone are not enough motivation to incorporate UI evaluations into the design process, consider some of the following reasons as well.

Scheduling evaluations during the design process provides useful feedback to the developers about how their design is perceived and how it can be improved. This creates a more iterative development process rather than the infamous build-and-fix model that can destroy timelines, budgets, and customer confidence. With useful feedback, developers can see their mistakes along the way and avoid them as they continue developing their application.

From expert reviews to surveys to psychologically oriented experiments, developers can use many different methods to create a comprehensive user interface evaluation plan. This flexibility allows for different kinds of software systems to have evaluation plans tailored specifically for each unique system. Using a variety of evaluations for a specific system allows developers to get feedback from sources such as user interface specialists,

productivity experts, novice computer users, advanced users, and even those with disabilities. All of this feedback allows for a more complete picture of how the software is perceived by the target audience. Surely this is an improvement over the system's designers alone trying to take an objective viewpoint when their personal contributions may come under scrutiny and must be evaluated fairly.

How are user interface designs evaluated?

Many techniques are available for evaluating user interfaces. Expert reviews, laboratory testing, surveys, acceptance tests, active evaluations and psychologically-oriented experiments are among the most well-known methods. These methods have a common goal of fairly judging an interface and providing the developers with useful feedback.

Expert reviews generally consist of a panel of less than five experienced and knowledgeable "experts" who will use several methods to evaluate a particular interface and provide a report with their findings. The experts may, for example, do a "cognitive walkthrough" where they will simulate common tasks that a user may execute. This process can also be done in an exploratory fashion where the experts may try various different paths through the software where they are simply browsing with no certain goal in mind. This type of evaluation can yield information about what a novice or experienced user may incur on a daily basis. This may expose some time-consuming procedures that could be reformed as well as inconsistencies throughout the system.

Laboratory testing usually entails a user demonstrating how they interact with the system while their movements and comments are recorded for future review. Reviewing the data collected from these experiments, developers can find areas where users became confused or had to make assumptions about the application that may not have been intended. Studying a user's interaction with a system can quickly reveal some areas of a design that need immediate attention as well as others which may be perfectly acceptable but could use clarification.

Surveys are a common and cheap way to gather feedback from a large number of users. As long as a survey is constructed with clear goals in mind, they can be very rewarding by showing a breakdown of responses for each part of a system. If a survey includes background information about the user, this information can be compared to how they evaluated the system and general trends can be found relating the user's demographic and their response to the survey. Surveys can quickly be deployed as a web form and statistical analysis can be generated with ease.

At the inception of a large software system, goals are created regarding the function and usability of the system upon completion. Acceptance tests are for verifying that these goals are met. Acceptance testing is most rewarding when clear and measurable criteria are stated to test. These criteria can be specified for different types of users based on age,

experience, language and other factors. Instead of finding flaws or areas to improve in the software based on user response, acceptance tests are for verifying how well a system has met its goals and objectives which were specified early in the design phase.

After a software system is deployed there is always room for improvement. Gathering impressions from frequent users and logging user activity and errors produced can be helpful to diagnose any areas which still may need improving. Directly contacting users through personal interviews has shown to be rewarding due to the helpful and constructive feedback that users often report. (166) Other means for passive feedback such as suggestion boxes, email support, and newsgroups allow users to receive answers for their problems and at the same time alert developers about areas or processes that may require some revision.

Lastly, controlled psychologically oriented experiments aim to define and measure properties of effective user interface designs. Using traditional scientific methods, hypothesis about an interface can be created, tested, and either confirmed or rejected. This can yield useful information which has the possibility of affecting more than just the interface in question. Perhaps new discoveries can be made about useful interface design and incorporated into future applications. Careful studies are made to ensure that any changes to the interface will yield an appreciable difference in user and market response.

What does this mean to developers?

If incorporating usability testing into the design and development plan of new software systems can lead to faster development, less cost, quicker adoption, and more market share, developers should take action. Even with all the benefits that can be found before releasing a software system, usability testing is an ongoing process aimed at continuous improvement and will result in a better designed system with measurably more satisfied users and more competition for effective user interface designs.

Evaluation of HousingMaps.com

Following the 8 Golden Rules of Shneiderman's "Designing the User Interface," the following evaluation will cover the popular housing-locating website that incorporated Google Maps and Craig's List.

1. Consistency
   The interface has few modes of operation – For Rent, For Sale, Rooms for Rent, Sublets. Each mode displays the relevant information in a consistent and recognizable manner. The table of listings found on the right side of the screen is organized the same for each mode. This design is consistent across all of its features and modes.

2. Shortcuts for frequent users
   The only such shortcut available is the ability to bookmark a link to a saved set of search criteria. This allows frequent users with the same search criteria to return and see current listings.

3. Informative feedback
   The website updates the map and listing view every time the search criteria are modified. The map will automatically re-center itself over a new area and the "flags" indicating search results will move accordingly. The map is easy to read and displays common landmarks such as freeways and city names.

4. Closure
   This site does not offer much sense of closure during or after a session of use. Some actions will divert your attention to another website for more details. This can be unexpected and unwanted when perusing search results.

5. Simple error handling
   Although there are few possibilities for error beyond a temporary loss of communication with the data sources, all errors are displayed in a clear and informative fashion.

6. Easy reversal of actions
   Since options are kept to a minimum, it is simple to revert to previous actions. The only visible drawback is when the user presses the "Back" button in their browser it does not reverse the last action, the user is directed to the last website they visited. The simple interface and lack of options makes the interface easy to navigate and encourages exploration.

7. Locus of control
   There is a strong feeling of interactive control over the system since it responds to your every action. Manipulating the search criteria and moving the map around the portal allows for extensive physical and psychological control over the website.

8. Short term memory load
   The overall simplicity of the website allows for a quick learning curve where the user is only required to be familiar with searching and manipulating a map. The website does not allow for an easy way to reference to a particular listing at any time in the future. Perhaps due to the volatile nature housing advertisements, this is not a significant problem.

HousingMaps.com incorporates many of the 8 Golden Rules although there are some areas which still need improvement. There is currently no way of returning to a previous search during a usage session without remembering the criteria and re-entering them.

The familiar analogy of a “back button” does not work with HousingMaps.com and may alienate the user by taking them to another site entirely. Also, the focus of the website can be directed to an external site when clicking on any of the housing listings. This could serve to confuse a user when their primary intention was to find the listing on the map. In summary, HousingMaps.com follows most of the important guidelines set forth in the 8 Golden Rules and stands as a web service that is expandable and familiar for the novice and advanced user.