An integrated method for evaluating interfaces

Heather L. McQuaid
MAYA Design
2100 Wharton Street
Suite 702
412-488-2900
mcquaid@maya.com

David Bishop
MAYA Design
2100 Wharton Street
Suite 702
412-488-2900
bishop@maya.com

ABSTRACT
To take advantage of the interdisciplinary experience of our colleagues, we decided several years ago to add heuristic evaluation to our expert analysis method. Although heuristic evaluation is a cost-effective method for evaluating interfaces, we found that the recommended prioritization strategy—ranking the problems according to severity—has several limitations. Specifically, it does not address how much it will cost the developers to fix the problems, nor does it adequately capture the distinction between high-level (global) and low-level (specific, screen-level) problems. To address these limitations, we developed a method which retains the richness of heuristic evaluation, but communicates the results in such a way that project managers, developers, and designers can form a clear and immediately executable plan for addressing the problems. Our method integrates user research, heuristic evaluation, affinity diagramming, cost-benefit charts, and recommendations into a report that others can use to plan both short and long-term improvements.

Keywords
Heuristic evaluation, usability inspection, affinity diagram, cost-benefit chart, return on investment (ROI).

BACKGROUND
MAYA is an interdisciplinary design consultancy that helps high-technology companies develop products that are functional, aesthetic and easy to use. We use a team-based approach to design [1], creating teams that include at least one member from each of our core disciplines, or departments: Human Sciences (Human Factors and Cognitive Psychology), Engineering (Software, Electrical, and Mechanical), and Design (Visual and Industrial). As part of our design process, we often evaluate existing products. One of the more cost-effective methods we use to evaluate products is an “expert analysis.”

As part of our expert analysis, we use heuristic evaluation
[6, 7]—a technique that allows usability specialists to use a list of heuristics, or guidelines, to evaluate products at virtually any point in the design process. However, we noticed several limitations with the technique. One limitation was that the prioritization strategy did not address the cost factor, that is, how much it would cost the client in both time and effort to fix the problems. Additionally, the prioritization strategy did not adequately capture the distinction between high-level problems (i.e., problems with navigation and task sequence) and low-level ones (i.e., problems with the usability of buttons, dialog boxes, and other elements that appear on a screen).

To address these concerns, we took heuristic evaluation and combined it with other techniques to create a repeatable method that produces a clear and immediately executable plan of attack for our clients.

OUR METHOD
Our method consists of five steps:

1. Gathering domain knowledge.
2. Conducting the heuristic evaluation.
3. Categorizing the issues.
4. Prioritizing the categories according to how important it is to fix them and how difficult it is to fix them.
5. Writing the report, including recommendations for solving the problems.

Gathering domain knowledge
As with any evaluation or design work, it’s imperative to understand the domain (i.e., users, tasks, environments, level of training needed) in which the product will be used. To gather this knowledge we may interview the client, analyze competitor’s products, and observe and interview users using techniques such as contextual inquiry [2] and task analysis [4, 5]. After gathering this information, we organize and analyze it using any number of techniques, including cultural models [2], information architectures [8], tasks flows [4, 5], user profiles, personas [3], and scenarios. We then use these analyses to provide context for evaluators as they conduct the heuristic evaluation.

Conducting the heuristic evaluation
After gaining an understanding of users and their tasks, we’re ready to conduct the heuristic evaluation. In general, we follow the instructions for heuristic evaluation originally outlined by Nielsen [5] and Nielsen and Molich [7]. Into these instructions we incorporate user personas and task lists.
Categorizing the problems
Because the number of problems identified during a heuristic evaluation can be daunting (typically between 100 and 200 issues), we use affinity diagramming to group the problems into manageable categories (generally between 10 to 15).

Prioritizing the problems
Once the problems are grouped into categories, we rank the categories according to how important it is from the users’ perspective to fix the problems in that group. Eventually all the categories are placed on the list, with the most important categories at the top and the less important at the bottom.

Next we take the numbered list of categories and place them on a chart according to how difficult it would be, from the developers’ and designers’ perspective, to fix the problems in the categories. Since the chart displays both the benefit to users of fixing the problems and the cost to the client of fixing the problems, we call it a Cost-Benefit Chart.

Cost-Benefit Chart
The figure below is an example of a Cost-Benefit Chart. The x-axis (horizontal) represents Importance (or benefit)—how important it is to fix the categories from the users’ perspective, that is, how much benefit users will experience if the problems are fixed. It is meant to be viewed as a continuum from lesser importance (!) to greater importance (!!).

The y-axis (vertical) represents Difficulty (or cost)—how much time, effort, and cost the client must expend to fix the problem. Likewise, it is meant to be viewed as a continuum from lesser difficulty ($) to greater difficulty ($$).

By mapping both Importance and Difficulty, the Cost-Benefit Chart essentially depicts the Return on Investment (ROI) associated with addressing each category of issues.

Creating the recommendations report
After we’ve categorized and prioritized the problems, we generate recommendations for fixing them and give the document to our clients. The report often opens their eyes to significant problems with their system or product. Rather than giving them a long list of individual problems, which might lead clients to believe they have a lot of busy-work to do, we point out that many of the individual problems are actually symptoms of a larger problem. By focusing on the larger problems, clients are able to think about architectural solutions rather than temporary remedies.

The document also fits most organizational cultures. People concerned with the big picture can use the Cost-Benefit Chart to get a high-level view of the problems, while those responsible for fixing specific problems can skip to the relevant section of the document to read our recommendations and look at the annotated screen shots.

ACKNOWLEDGMENTS
We thank Steve Little, Mark Lotter, and Chris Holly for comments on earlier drafts.

REFERENCES
5. Lewis, C., and Rieman, J. Task-centered User Interface Design: A Practical Introduction. Available at: http://www.acm.org/~perlman/uidesign.html#chap0