

## **Design Principles and Usability Heuristics**

**Heuristic Evaluations: An “introspective” method that can be used to inspect an interface for usability problems.**

**The heuristics are also good to keep in mind when trying to avoid common design pitfalls...**

### **Design principles and usability heuristics (I)**

**The design principles represent a broad set of general rules based on research and experience that also describe features of “usable” systems.**

- broad usability statements that guide a developer’s design efforts
- derived by evaluating common design problems across many systems

**Heuristic evaluation: Take these same principles and use them to “evaluate” a system for usability problems.**

- Reasonably popular approach since actual user involvement is not required (cheaper and logistically easier) and it end up catching many design flaws.
- Is considered an “expert review” technique.

## Design principles and usability heuristics (II)

### Advantages

- It is a “minimalist” approach where using a few general guidelines grounded in research and experience can help identify and correct the majority of usability problems.
  - Also, reasonable easily-remembered list and easily applied with modest effort.
- Considered “discount usability engineering” due to its relative low cost and the speed at which it can be applied. However, the experience level of the evaluators has an impact, so often done by usability experts.

## Design principles and usability heuristics (III)

### Challenges (for lack of a better word)

- These principles can’t be treated as a simple checklist.
  - Note: “If done wrong, that’s bad” is a common “disadvantage”, but it is worth noting here.
- There are subtleties involved in their use and in classifying some specific issues that are raised.
- Some consider this a stage before “real” user testing to catch many issues before users are brought in for usability/performance testing

## Why is “discount usability engineering” approach?

**Relative to user-observational studies, this can be cheap and fast and relatively easy for trained practitioners, which can be critical in today’s product cycle...**

- There are no special labs or equipment needed.
  - For many things, likely able to run it on your own machine in your office
  - Interesting bonus: can even be used on *paper prototypes*
- Doing this type of evaluation can be done on the order of one day where other usability testing could take weeks.
- Once the approach is understood by a team it can be used in many scenarios with little additional learning and the more careful you are, the better the results get.

Evan Golub / Ben Bederson / Saul Greenberg

## Heuristic Evaluation

**Developed by Jakob Nielsen (1990)**

- Original list of heuristics seems inspired by Shneiderman’s “Eight Golden Rules” of design.
- Nielsen has had multiple similar lists over the years.
- Jill Gerhardt-Powals has a list as well but they have a very different feel to them.

**Helps find usability problems in a UI design**

**Small set (3-5) of evaluators examine UI**

- Independently check for compliance with usability principles
- Evaluators only communicate after they are done with their eval and the findings are then aggregated.
- Common for overlap but in places different evaluators will find or identify different problems

Evan Golub / Ben Bederson / Saul Greenberg

## Heuristic Evaluation Process

### **Evaluators go through UI several times**

- inspects various dialogue elements
- compares with list of usability principles
- consider other principles/results that come to mind

### **Usability principles**

- Nielsen's "heuristics"
  - there are several slightly different sets (we will see one) of heuristics
- supplementary list of category-specific heuristics
  - competitive analysis & user testing of existing products

### **Use violations to redesign/fix problems**

Evan Golub / Ben Bederson / Saul Greenberg

## Phases of Heuristic Evaluation

### **1) Pre-evaluation training**

- give evaluators needed domain knowledge and information on the overall scenario/context

### **2) Evaluation**

- individuals on the evaluation team review the design of the prototype or system and then come together to aggregate their results

### **3) Severity rating**

- individually and then as a group determine how severe each problem is (priority)

### **4) Debriefing**

- discuss the outcome of the evaluation with design team

## **How to Perform Evaluation**

### **At least two passes for each evaluator**

- first to get feel for flow and scope of system
- second to focus on specific elements

### **If system is walk-up-and-use or evaluators are domain experts, then no assistance needed**

- otherwise design team might supply evaluators with scenarios

### **Each evaluator produces list of problems**

- explain why each is a problem, with reference to heuristic(s) or other info
- be specific and list each problem separately

Evan Golub / Ben Bederson / Saul Greenberg

## **Design Principles and Usability Heuristics**

- 1: Simple and natural dialogue**
- 2: Speak the users' language**
- 3: Minimize user's memory load**
- 4: Be consistent**
- 5: Provide feedback**
- 6. Provide clearly marked exits**
- 7. Provide shortcuts**
- 8: Deal with errors in a positive and helpful manner**
- 9. Provide help**

**Note: This is not the only list teams use, but it's the one on which we will focus.**

## Severity Rating

**Used to allocate resources to fix problems**

**Estimates of need for more usability efforts**

**Combination of**

- frequency
- impact
- persistence (one time or repeating)

**Should be done independently by all evaluators and then aggregated to a final calculation after all evaluations are in.**

Evan Golub / Ben Bederson / Saul Greenberg

## Nielsen's Example Ratings List

**0 = I don't agree that this is a usability problem at all.**

**1 = Cosmetic problem only.**

**need not be fixed unless extra time is available on project**

**2 = Minor usability problem.**

**fixing this should be given low priority**

**3 = Major usability problem.**

**important to fix, so should be given high priority**

**4 = Usability catastrophe.**

**imperative to fix this before product can be released**

### Some comments on the above...

- Although Nielsen provides a "0" rating, it is unclear where it would be used
  - perhaps on a "second opinion" evaluation
- It is possible for a cosmetic problem to be a usability catastrophe
  - imagine a green checkmark meaning "bad/danger"

## Examples of individual entries

### **Can't copy info from one window to another.**

violates "Minimize the users' memory load"

severity: (3) major

fix: allow copying

### **Typography uses mix of upper/lower case formats and fonts.**

violates "Consistency and standards"

slows users down

probably wouldn't be found by standard user testing

severity: (1) cosmetic

fix: pick a single format for entire interface

### **Green flashing lights mean system settings are being changed, red lights means normal functionality taking place.**

violates "Consistency and standards"

could confuse new users

severity: (2) minor issue, cosmetic fix

fix: reverse color usage of lights

## Debriefing

**Conduct with evaluators, observers, and design/development team members**

**Discuss general characteristics of UI**

**Suggest potential improvements to address major usability problems**

**Development team rates how hard things are to fix**

**Suggestion: Make it a brainstorming session...**

- tend to try to avoid direct criticism until end of session

## Results of Using HE

### **Discount: benefit-cost ratio of 48 [Nielsen94]**

- cost was \$10,500 for benefit of \$500,000
- value of each problem ~15K (Nielsen & Landauer)
- how might we calculate this value?
  - in-house → productivity
  - open market → sales

### **Correlation between severity & finding w/ HE**

[http://www.useit.com/papers/heuristic/heuristic\\_evaluation.html](http://www.useit.com/papers/heuristic/heuristic_evaluation.html)

Evan Golub / Ben Bederson / Saul Greenberg

## Why Multiple Evaluators?

### **Single evaluator achieves poor results**

Previous comparisons indicated that a single evaluator will only find around 35% of the actual usability problems but that using five evaluators will lead to finding around 75% of them (some don't get found until the users try it out).

One question that came up was “why not more evaluators?” Would it help to go up to 10? 20?

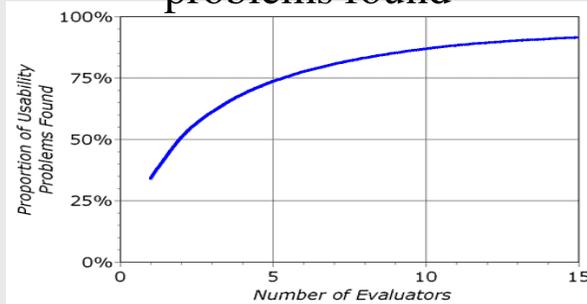
- The reality is that adding evaluators costs more (not just scaling for number of people but also increased time for everyone during the aggregation stage).
- Having that many evaluators won't identify many more problems in practice.

Evan Golub / Ben Bederson / Saul Greenberg

## Why Multiple Evaluators (cont)?

(graphs for a specific example study that was done)

problems found



benefits / cost



Evan Golub / Ben Bederson / Saul Greenberg