Design Principles and Usability Heuristics - Part I

You can avoid common design pitfalls by following 9 design principles

You can inspect an interface for usability problems with these principles

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

Broad “rules of thumb” that describe features of “usable” systems

**Design principles**

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

**Heuristic evaluation**

- same principles used to “evaluate” a system for usability problems
- becoming very popular
  - user involvement not required
  - catches many design flaws
- is an “expert review”

Evan Golub / Ben Bederson / Saul Greenberg
Design principles and usability heuristics (II)

Advantages
• the “minimalist” approach
  – a few general guidelines can correct for the majority of usability problems
  – easily remembered, easily applied with modest effort

• discount usability engineering
  – cheap and fast way to inspect a system
  – can be done by usability experts

Disadvantages:
• 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.
• subtleties involved in their use

Discount Usability Engineering
Cheap/Fast/Easy To Use!
• no special labs or equipment needed
  – might even be able to run it on your own machine in your office
  – can even be used on paper prototypes

• can be on order of 1 day to apply
  – standard usability testing may take weeks

• once understood, can use in many scenarios with little additional learning

• the more careful you are, the better it get
**Heuristic Evaluation**

Developed by Jakob Nielsen (1990)
- seems inspired by Shneiderman’s “Eight Golden Rules”

Helps find usability problems in a UI design

Small set (3-5) of evaluators examine UI
- independently check for compliance with usability principles (“heuristics”)
- different evaluators will find different problems
- evaluators only communicate afterwards
  - findings are then aggregated

Can perform on working UI or on sketches

**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
Phases of Heuristic Evaluation

1) Pre-evaluation training
   • give evaluators needed domain knowledge and information on the scenario

2) Evaluation
   • individuals evaluate and then aggregate results

3) Severity rating
   • determine how severe each problem is (priority)

4) Debriefing
   • discuss the outcome 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 might supply evaluators with scenarios

Each evaluator produces list of problems
   • explain why with reference to heuristic or other info.
   • be specific and list each problem separately
Examples

Can’t copy info from one window to another
• violates “Minimize the users’ memory load”
• 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 user testing
• fix: pick a single format for entire interface

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 calculated after all evaluations are in

Should be done independently by all judges
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”

Debriefing

Conduct with evaluators, observers, and 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

Make it a brainstorming session
• little 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


Why Multiple Evaluators?

Single evaluator achieves poor results
- only finds 35% of usability problems
- 5 evaluators find ~ 75% of usability problems
- why not more evaluators???? 10? 20?
  - adding evaluators costs more
  - many evaluators won’t find many more problems
Why Multiple Evaluators (cont)?

problems found

benefits / cost

(Graphs for a specific example)

Recap: Three Evaluation Methods

1. Usability analysis with users (previously discussed)
   - Interviews
   - Questionnaires
   - Silent observation
   - Think aloud
   - Constructive interaction

   • Discount methods such as Heuristic Evaluation
     - Discussion of the technique (this set of notes)
     - 9 heuristics (associated set of notes)

   • Controlled studies (touched on previously, continued soon)