Why Heuristic Evaluation?

- What are Heuristics?
Why heuristic evaluation?

- What are Heuristics?

- Why Heuristic Evaluation?
  - Making the ROI case for usability
  - Showing that even minimal systematic effort better than winging it

Heuristic Evaluation Overview

- Developed by Jakob Nielsen in late 80’s-early 90’s
  - Focused on ROI and simplicity
  - Trying to convince world usability was worth paying attention to
  - A simple structured process for finding usability problems in a UI
  - Small set (3-5) of evaluators examine UI
  - Can perform on working UI or on sketches
Heuristic Evaluation Process

• Evaluators go through UI several times independently
  – Inspect dialogue elements
  – compare with list of usability principles
  – consider other principles/results that come to mind
  – compare results afterwards to compile master list

• Usability principles
  – Nielsen’s “heuristics”
  – supplementary list of category-specific heuristics
    • competitive analysis & user testing of existing products

Key principles

• Short set of heuristics
  – Simplicity is key
  – Evaluator should be able to keep all in his head

• Meant to be performed on actual UI
  – Not meant as a thought experiment

• Evaluators must work independently

• Only minimal training necessary for evaluators
Evaluation

- Nielsen & Molich did a set of studies in 1990
- Examined vast set of common usability problems in a wide variety of applications
- From a set of 249, they identified 10 common basic principles
- They did a study where they trained college students to do the analysis, and compared their performance on a system where issues were known

Why Multiple Evaluators?

- Every evaluator cannot find every problem
- Good evaluators find both easy & hard problems
- Easy to get fixated on specific issues
Decreasing Returns

problems found

![Graph showing decreasing returns on problems found with increasing evaluators.]

benefits / cost

![Graph showing benefits to cost ratio with increasing evaluators.]

Still very cost effective

Instrumental to launching UE as a field

10 Heuristics

1. **Visibility of system status** The system should keep users informed through appropriate feedback within reasonable time.

2. **Match between system and the real world** The system should speak the users' language. Follow real-world conventions, making information appear in a natural and logical order.

3. **User control and freedom** Users often choose system functions by mistake and will need a clearly marked "emergency exit". Support undo and redo.

4. **Consistency and standards** Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

5. **Error prevention** Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
10 Heuristics

6. **Recognition rather than recall**
   Minimize memory load - make objects, actions, and options visible. User should not have to remember info between screens. Instructions should be visible.

7. **Flexibility and efficiency of use**
   Accelerators -- unseen by the novice user – may speed up the interaction for expert users. Allow users to tailor frequent actions.

8. **Aesthetic and minimalist design**
   Dialogues should not contain info which is irrelevant or rarely needed. Every unit of information diminishes relative visibility.

9. **Help users recognize, diagnose, and recover from errors**
   Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

10. **Help and documentation**
    Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Is this all there is to good UI design?
HE vs. User Testing

• HE is much faster
  – 1-2 hours each evaluator vs. days-weeks

• HE doesn’t require interpreting user’s actions

• User testing is far more accurate (by def.)
  – takes into account actual users and tasks
  – HE may miss problems & find “false positives”

• Good to alternate between HE & user testing
  – find different types of problems
  – don’t waste participants’ time

Results of Using HE (cont.)

• Single evaluator achieves poor results
  – only finds 35% of usability problems
  – 5 evaluators find ~ 75% of usability problems
  – why not more evaluators???? 10? 20?
Results of Using HE (cont.)

- 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
- What about expertise?