Heuristic Evaluation

Prof. James A. Landay & Prof. Ge Wang
Computer Science & Music Departments
Stanford Center at Peking University
Summer 2016

Heuristic Evaluation

Developed by Jakob Nielsen

Helps find usability problems in a UI design

Small set (3-5) of evaluators examine UI
- independently check for compliance with usability principles ("heuristics")
- evaluators only communicate afterwards
  - findings are then aggregated
  - use violations to redesign/fix problems

Can perform on working UI or on sketches

Evaluation

About figuring out how to improve design
Issues with lo-fi tests?

Not realistic
- visuals & performance
Not on actual interface
- can’t test alone
Need participants
- can be hard to find repeatedly

Heuristic Evaluation

Heuristic Evaluation Overview
The Heuristics
Exercise

Unsuccessful
Evaluators
Successful
Usability Problems
Hard
Easy
Heuristics

H2-1: Visibility of system status
H2-2: Match between system & real world
H2-3: User control & freedom

Heuristics (cont.)

H2-4: Consistency & standards
H2-5: Error prevention
H2-6: Recognition rather than recall

H2-7: Flexibility and efficiency of use
H2-8: Aesthetic & minimalist design
H2-9: Help users recognize, diagnose, & recover from errors

Good Error Messages

- Clearly indicate what has gone wrong
- Human readable
- Polite
- Describe the problem
- Explain how to fix it
- Highly noticeable

Heuristic Violation Examples

1. [H1-3 Minimize the users’ memory load] Can’t copy info from one window to another
   - fix: allow copying

2. [H2-4 Consistency and Standards] Typography uses different fonts in 3 dialog boxes
   - slows users down
   - probably wouldn’t be found by user testing
   - fix: pick a single format for entire interface
Severity Ratings

0 - don’t agree that this is a usability problem
1 - cosmetic problem
2 - minor usability problem
3 - major usability problem; important to fix
4 - usability catastrophe; imperative to fix

Severity Ratings Example

1. [H1-4 Consistency] [Severity 3]
The interface used the string “Save” on the first screen for saving the user’s settings, but used the string “Store” on the second screen. Users may be confused by this different terminology for the same function.

Decreasing Returns

<table>
<thead>
<tr>
<th>problems found</th>
<th>benefits / cost</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="Graph 1" /></td>
<td><img src="#" alt="Graph 2" /></td>
</tr>
</tbody>
</table>

* Caveat: graphs for a specific example

Heuristic Evaluation Summary

- Have evaluators go through the UI twice
- Ask them to see if it complies with heuristics
  - note where it doesn’t & say why
- Combine the findings from 3 to 5 evaluators
- Have evaluators independently rate severity
- Alternate with user testing
Find 12-15 Heuristic Violations

1. H2-4 Consistency; remove column, 4th item is different w/ checkboxes. [150]
2. H2-9 Error prevention; non-numeric data in the quantity. Do not allow. [125]
3. H2-2 Match between system & real world; vehicle selection link not language I’d expect [100]
4. H2-1 Visibility of System Status; unclear which item to remove based on error message (“red/bold”). [150]

Further Reading

- Longer lecture
  - https://drive.google.com/file/d/0BweiB6wu4sBNZ2tZGxKb2tOTg/view
- Books
  - Usability Engineering, by Nielsen, 1994
- Web site
  - http://www.nngroup.com/articles/

Next Time

- Lecture
  - Heuristic Evaluation Group Exercise
- Read
  - How to Conduct a Heuristic Evaluation by Jakob Nielsen
- Project
  - Medium-fi Prototype & 3 Tasks due on Sunday 6 PM
  - Sunday night you will do Heuristic Evaluation of other team (by yourself – write up report)