Heuristic Evaluation

“Discount” Usability Testing

Evaluating UI Designs

• Usability testing is a major technique
  – Formal techniques require users, rigid control experiments, statistical analysis
• “Discount” methods don’t require users
  – Heuristic Evaluation
  – Cognitive Walkthrough

Adapted from material by Marti Hearst, Loren Terveen
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")
  - different evaluators will find different problems
  - evaluators only communicate afterwards
    - findings are then aggregated
- Can perform on working UI or on sketches

Phases of Heuristic Evaluation

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

Adapted from slide by James Landay
Jakob Nielsen’s heuristics

<table>
<thead>
<tr>
<th>1.0 – circa 1990</th>
<th>2.0 – circa 1994</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1. Simple and natural dialog</td>
<td>Aesthetic and minimalist design</td>
</tr>
<tr>
<td>H2. Speak the user’s language</td>
<td>Match between system and real world</td>
</tr>
<tr>
<td>H3. Minimize user memory load</td>
<td>Recognition rather than recall</td>
</tr>
<tr>
<td>H4. Be consistent</td>
<td>Consistency and standards</td>
</tr>
<tr>
<td>H5. Provide feedback</td>
<td>Visibility of system status</td>
</tr>
<tr>
<td>H6. Provide clearly marked exits</td>
<td>User control and freedom</td>
</tr>
<tr>
<td>H7. Provide shortcuts</td>
<td>Flexibility and efficiency of use</td>
</tr>
<tr>
<td>H8. Provide good error messages</td>
<td>Help users recognize, diagnose, and recover from errors</td>
</tr>
<tr>
<td>H9. Prevent errors</td>
<td>Error prevention</td>
</tr>
<tr>
<td>H10. Help and documentation</td>
<td>Help and documentation</td>
</tr>
</tbody>
</table>

Pros / Cons

- + Cheap (no special lab or equipment)
- + Easy
- + Fast (about 1 day)
- + Cost-effective
- + Detects many problems without users
- + Complementary to task-centered approaches
- + Coverage
- + Catches cross-task interactions
- - Requires subjective interpretation
- - Does not specify how to fix problems
- - Performance improves as evaluator knowledge increases
Procedure

• A set of evaluators (3-5 is about optimal) evaluate a UI (some training may be needed)
• Each one independently checks for compliance with the heuristics
  – Different evaluators find different problems
• Evaluators then get together and merge their findings
• Collectively rate severity of the problems
• Debriefing/brainstorming → how to fix the problems (and point out what’s really good)

How to Perform H.E.

• At least two passes for each evaluator
  – first to get feel for flow and scope of system
  – second to focus on specific elements
• Assistance from implementors/domain experts
  – If system is walk-up-and-use or evaluators are domain experts, then no assistance needed
  – Otherwise might supply evaluators with scenarios and have implementors standing by

Adapted from slide by James Landay
How to Perform Evaluation

- Where problems may be found
  - single location in UI
  - two or more locations that need to be compared
  - problem with overall structure of UI
  - something that is missing

Example Problem Descriptions

- Have to remember command codes
  - Violates “Minimize the users’ memory load” (H3)
  - Fix: add drop down box with selectable codes
- Typography uses mix of upper/lower case formats and fonts
  - Violates “Consistency and standards” (H4)
  - Slows users down
  - Probably wouldn't be found by user testing
  - Fix: pick a single format for entire interface
Severity ratings

- Used to allocate resources to fix problems
- Should be calculated after all evaluations are done
- Should be done independently by all evaluators
- Based on
  - Frequency the problem will occur
  - Impact of problem (hard or easy to overcome)
  - Persistence (will users learn a work around or will they be bothered every time?)
- 1 – cosmetic problem
- 2 – minor usability problem
- 3 – major usability problem; important to fix
- 4 – usability catastrophe – must fix

Severity Ratings Example

1. [H4 Consistency] [Severity 3]

The interface used the string "Save" on the first screen for saving the user's file, but used the string "Write file" on the second screen. Users may be confused by this different terminology for the same function.
Debriefing

- Conduct with evaluators, observers, and development team members
- Discuss general characteristics of UI
- Suggest potential improvements to address major usability problems
- Developer team rates how hard things are to fix
- Make it a brainstorming session

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
    - adding more evaluators doesn’t increase the number of unique problems found
Decreasing Returns

- (from Nielsen)
- Caveat: these graphs are for a specific example
- This is a controversial point.

Adapted from slide by James Landay

Why Multiple Evaluators?

- Every evaluator doesn’t find every problem
- Good evaluators find both easy & hard ones
Exercise

• Evaluate an application using heuristic evaluation
  – Bring your computer if you have one!
  – Refer back to slide with the 10 heuristics
• Fill out issues found on next slide(s) and submit them when you are done or we run out of time, whichever comes first
• We will discuss and debrief after you are done

Heuristic Evaluation Issue Log
Example

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>H4-Consistency</td>
<td>3</td>
<td>The interface used the string &quot;Save&quot; on the first screen for saving the user's file, but used the string &quot;Write file&quot; on the second screen. Users may be confused by this different terminology for the same function</td>
</tr>
<tr>
<td>H8-Error Message</td>
<td>4</td>
<td>Entering invalid input into dialog box on first form results in “Error 103”</td>
</tr>
</tbody>
</table>

...
### Heuristic Evaluation Issue Log

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heuristic</td>
<td>Severity</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>

Heuristic Evaluation Issue Log