Usability Engineering
Evaluation Methods

Matthias Rauterberg
HG, 3.53
TU/e - ID - DI
g.w.m.rauterberg@tue.nl
Website:
http://www.idemployee.id.tue.nl/g.w.m.rauterberg/

Content today

- Analytical or Inspection methods
- Heuristic Evaluation procedure
- Heuristic Evaluation discussion
- Alternative heuristics
- Cognitive Walkthrough (CW) procedure
- CW discussion
Analytical evaluation methods

- Emphasis: predict usability problems!

- Methods differ in whether it is based on theory, what theory it is based on and thus in scope and applicability

Overview over Evaluation methods
Inspection methods
(Nielsen and Mack, 1994):

- Evaluators examine usability-related aspects of user interface
- Use of inspection problem reports:
  - Leads to fixes and redesign suggestions
  - Prioritised list of usability problems, based on severity of problems
  - Software cost associated with implementing suggested redesign

Inspection methods

- Heuristic evaluation
- Guideline review
- Pluralistic walkthrough
- Consistency inspections
- Standards inspections
- Cognitive walkthroughs
- Formal usability inspections
Differences

- Number and detail of heuristics/guidelines
- Amount of task/goal focus
  - specific tasks
  - open-ended examination
- Individual versus group inspections
  - multiple views, multiple expertise
- Ease of use and ease of learning

Validity of inspection results

- How predictive are the results of end-user problems?
- At most predict 30-50% of problems found with user testing
Effectiveness of inspection results

- Effectiveness in improving usability of product
- Need for clear problem reports: e.g.
  - What is the problem
  - What is the cause
  - What interface component is involved
  - Severity of problem
  - Ideas for improvements

Trade-offs

- Global evaluation versus detailed evaluation: heuristics versus many guidelines
- Task-oriented versus open-ended evaluation
- Based on detailed user profiles, or less well-defined user profiles
- Consideration of number of design trade-offs
Heuristic Evaluation (HE)

- Limited set of heuristics
- Scope:
  - Source of heuristics
  - Task oriented approach
  - Professional domain

(Similar to Shneiderman’s 8 golden rules)

Heuristic Evaluation (HE)

- Compare design against principles [P-x]
- For example:
  - [P-1] Error prevention
  - [P-2] Flexibility and efficiency of use
  - [P-3] Help users recognize, diagnose, and recover from errors
  - ...

[P-1] Prevent errors

- Disabled functions should appear as such.

![Disabled function example in Microsoft Word](image)

This command is not available because a document window is not active.

[P-2] Flexibility and frequency of use

- Accelerators speed up the interaction for the expert user
- Allow users to tailor frequent actions.
- Example: incorrect assumptions

![Example of incorrect assumptions](image)
[P-3] Help users recognize and recover from errors

- Error messages should be expressed in plain language

![Error message example](image)

[P-4] Recognition rather than recall

- Provide only limited number of options

![Option example](image)
How to measure [P-5] and [P-6]

**[visual] feedback (FB)**

Interactive directness (ID)

<table>
<thead>
<tr>
<th>Low</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch</td>
<td>Menu interface</td>
</tr>
<tr>
<td>Command language</td>
<td>Direct manipulation</td>
</tr>
</tbody>
</table>

**[functional] feedback**

\[

d_{FB} = \frac{1}{D} \sum_{d=1}^{D} \left( \frac{PF_d}{HF_d} \right) \times 100\%
\]

**Interactive directness**

\[

ID = \left\{ \frac{1}{P} \sum_{p=1}^{P} \log(\text{PATHp}) \right\}^{-1} \times 100\%
\]
## Standards and Norms: overview

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>suitability for the task</td>
<td>suitability (activity adapted)</td>
<td>suitability for the task</td>
<td>task orientation</td>
</tr>
<tr>
<td>self-descriptiveness</td>
<td>feedback about system states</td>
<td>self-descriptiveness</td>
<td>transparency</td>
</tr>
<tr>
<td>conformity with user</td>
<td>appropriate format and pace of</td>
<td>conformity with user</td>
<td>feedback</td>
</tr>
<tr>
<td>expectations</td>
<td>information presentation</td>
<td>expectations</td>
<td></td>
</tr>
<tr>
<td></td>
<td>information and instruction of</td>
<td></td>
<td>compatibility</td>
</tr>
<tr>
<td></td>
<td>user</td>
<td></td>
<td>consistency</td>
</tr>
<tr>
<td></td>
<td>ease of use applicable to skill</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>level</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>hearing and participation of</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>users</td>
<td></td>
<td></td>
</tr>
<tr>
<td>controllability</td>
<td></td>
<td>controlability</td>
<td>support</td>
</tr>
<tr>
<td>error robustness</td>
<td></td>
<td>error tolerance</td>
<td>selection possibilities</td>
</tr>
</tbody>
</table>

### Procedure for finding problems and variants

- **Training session of product**
- **Actual evaluation:**
  - two passes using a scenario:
    1. get a feel for interface,
    2. focus on heuristics
- **Debriefing session for discussion of outcomes**
- **Severity rating of problems**
The evaluator effect with heuristic evaluation

- Validity of method:
  same set of problems, with other methods

- Different evaluators find different problems

- What kind of expertise is required?
  - Usability in general
  - Domain expertise: e.g. data entry tasks
  => advise on optimum number of evaluators

Which evaluators found which problems?
How many evaluators found how many problems?

Heuristic Evaluation: pros and cons

- **Good points:**
  - useful as a checklist when developing UI
  - useful for ‘quick and dirty’ evaluation
    (see Nielsen’s ‘discount usability engineering’ approach)

- **Not so strong points**
  - the rules are a mixed bag (no underlying theory)
  - sometimes they are mutually contradictory
  - ambiguous, hard to interpret
Severity of a usability problem

- The frequency with which the problem occurs: 
  *Is it common or rare?*

- The impact of the problem if it occurs: 
  *Will it be easy or difficult for the users to overcome?*

- The persistence of the problem: 
  *Is it a one-time or a recurring problem?*

Severity Rating [0-4]:

0 Don’t agree this is a usability problem
1 Cosmetic problem only – need not be fixed unless time is available
2 Minor usability problem – fixing this should be given low priority
3 Major usability problem – important to fix, high priority
4 Usability catastrophe – imperative to fix
Alternative Heuristics:  
(Malone and Lepper 1987)

- Background: intrinsically motivating educational environments
- Emphasis on better learning, because it is fun! 
  ⇒ different set of heuristics
- Furthermore, see website: 
The interactive heuristic evaluation toolkit 
(with different sets of heuristics):  

Malone and Lepper’s heuristics

- Developed in the context of Educational Environments and Games
- Emphasis on motivation to learn, because of intrinsic motivation (e.g. fun)
- Assumption:
  - Providing intrinsic motivation, influences what and how people learn (and possibly perceive interaction in general)  
Heuristics for Intrinsically Motivating Instructional Environments

- Individual motivations
  - Challenge (appropriate level of difficulty)
  - Curiosity (appropriate level of informational complexity)
  - Control (user should feel in control)
  - Fantasy (fantasy should be appealing)
- Interpersonal motivations
  - Cooperation
  - Competition
  - Recognition

Example: computer game

- Flight simulator

How to interpret:
  - Challenge
  - Fantasy
  - Control
Challenge: The Mooney IFR panel.

Fantasy: San Francisco's Golden Gate Bridge at sunrise, as seen from the Cessna 182RG.
Control: example

This is the Bell 206

The Sopwith flying over mountains.

Cognitive Walkthrough (CW)

- Predicts usability problems
- Focus on one aspect of usability: ease of learning!
  - Based on theory of learning by exploration
- Evaluates interface in the context of one or more specific tasks.
- Done by individuals or by groups.
Theoretical background

- Focus on novice explorative learning of computer interfaces

- Problem solving theories

- Theoretical model of learning by exploration (CE+ theory from Polson & Lewis 1990)

Components of CW

- Problem solving component
  - when faced with a set of untried actions: CE+ chooses action with most overlap with goal

- Learning component
  - evaluates whether feedback contains terms of the user goals, and stores the outcome of the action

- Execution component
  - executes rules and coordinates execution of rules
Example CW: “Task: open new file”

- **Problem solving:**
  - Check visible menu names for overlapping terms (File)
  - Select menu name with most overlap

- **Learning component**
  - Assess feedback, store results: “New blank document”

- **Execution component**
  - Decides whether to do problem solving, or applies knowledge through learning component

Assess overlap with goal (1)
Assess overlap with goal (2)

Assess overlap with goal (3)
Evaluate results and store

Usability principles based on CE+

- Make available actions visible
- Use identity cues between actions and goals
- Use identity cues between feedback and goals
- Provide obvious way to undo actions
- Make available actions easy to discriminate
- Offer few alternatives
- Tolerate at most one hard-to-understand action in a repertoire
- Require as few choices as possible
Input to the CW:

- Interface's detailed design description (paper mock-up or working prototype)
- Task scenario
- Assumptions about the user group
- Context of use
- Sequence of actions, to complete each task

CW Analyst asks four questions:

- Will the user try to achieve the right effect?
- Will the user notice that the correct action is available?
- Will the user associate the correct action with the effect to be achieved?
- Will the user see that progress has been made?
Criticisms to CW

- Tedious to use:
  - answers many questions for each action
- Limited scope: ease of learning
  - Misses consistency problems, general problems and recurring problems
- Only finds problems to correct action sequences, not recovery-from-error problems
- Difficult to apply without knowledge about background theory

The evaluator effect in CW
(Hertzum and Jacobsen 1998)

- Large differences between evaluators in number of problems found
- What is found is influenced:
  - by assumptions about users (how diverse)
  - by anchoring to evaluator’s own experience
Evaluator effect in CW
(Hertzum and Jacobsen 1998)

- Who found which problems?
  Each row represents an evaluator, each column a problem.
Literature for next lecture

- Dumas and Redish, 1999, chapters 7, 11, 12, 18

- Boren and Ramey, 2001
  Thinking Aloud
  (thinkaloud[2001].pdf on 0H420 website)

References


  - Chapter 1: Executive summary, 1-24.
  - Chapter 5: The cognitive walkthrough method: a practitioner’s guide, 105-140.