Übung zur Vorlesung

Mensch–Maschine–Interaktion 1

Aurelien Tabard
Ludwig–Maximilians–Universität München
Sommersemester 2012
Evaluate your project
Iterative design

Observation

evaluation

implementation

design
Heuristic Evaluation

Heuristic evaluation is a “discount” usability inspection method

- Quick, cheap and easy evaluation of UI design

Implicit assumptions:

- There is a fixed list of desirable properties of user interfaces (the “heuristics”)
- These heuristics can be checked by experts with a clear and defined result
Ten Usability Heuristics

- Meet expectations
  1. Match the real world
  2. Consistency & standards
  3. Help & documentation

- User is boss
  4. User control & freedom
  5. Visibility of system status
  6. Flexibility & efficiency

- Errors
  7. Error prevention
  8. Recognition, not recall
  9. Error reporting, diagnosis, and recovery

- Keep it simple
  10. Aesthetic & minimalist design

http://www.useit.com/jakob/photos/
Procedure

- Small set of evaluators examine the interface and judge its compliance with recognized usability principles (the "heuristics").
- Either just by inspection or by scenario-based walkthrough
- Critical issues list, weighted by severity grade
- Opinions of evaluators are consolidated into one report
Number of evaluators

Every evaluator doesn’t find every problem
Good evaluators find both easy & hard ones
Number of evaluators

Single evaluator achieves poor results
Only finds 35% of usability problems
5 evaluators find ~ 75% of usability problems
Heuristics

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation
Heuristics

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation
Heuristic

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation
Heuristic

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation
Heuristic

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation
Heuristic

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation
Heuristic

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- **Flexibility and efficiency of use**
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation
Heuristic

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation
Heuristic

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation
Heuristic

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation
Severity scale

Contributing factors

‣ Frequency: how common?
‣ Impact: how hard to overcome?
‣ Persistence: how often to overcome?

Severity scale

‣ Cosmetic: need not be fixed
‣ Minor: needs fixing but low priority
‣ Major: needs fixing and high priority
‣ Catastrophic: imperative to fix
Writing good heuristic evaluations

- Heuristic evaluations must communicate well to developers and managers
- Include positive comments as well as criticisms
  - Good: Toolbar icons are simple, with good contrast and few colors (minimalist design)
- Be tactful
  - Not: the menu organization is a complete mess
  - Better: menus are not organized by function
- Be specific
  - Not: text is unreadable
  - Better: text is too small, and has poor contrast (black text on dark green background)
Example

What to include:

- Problem
- Heuristic
- Description
- Severity
- Recommendation (if any)
- Screenshot (if helpful)

Severe: User may close window without saving data (error prevention)

If the user has made changes without saving, and then closes the window using the Close button, rather than File >> Exit, no confirmation dialog appears.

Recommendation: show a confirmation dialog or save automatically
Summary

› Heuristic evaluation is a discount method
› Have evaluators go through the UI twice
  • Ask them to see if it complies with heuristics
  • Note where it doesn’t and say why
› Have evaluators independently rate severity
› Combine the findings from 3 to 5 evaluators
› Discuss problems with design team
› Cheaper alternative to user testing
› Finds different problems, so good to alternate