Evaluation without users

Readings: Dix 9.1, 9.2, 9.3
Announcements

- Assignment 3 will be about the heuristic evaluation and cognitive walkthrough of your project.
- Will be done in groups.
- Replaces project report
- Weight: 9%
- New marking scheme for the project:
  - 5% short oral presentations
  - 5% team ranking
  - 10% final demo + presentation: July 24, 27; deadline July 24
  - 10% final UI interface (Installation CD): deadline July 24
  - Testing with children (optional) will be organized by Robyn Hovorka. You may include results from user testing in Assignment 3.
Introduction

- Evaluation is easier than design.

- The principles that drive design, drive evaluation as well.

- Evaluation is iterative and must be applied to all design iterations (lo-fi to hi-fi prototypes and final product)
Heuristic evaluation

- performed by a usability expert
- Steps
  - Inspect UI thoroughly
  - Compare UI against heuristics
  - List usability problems
    - Explain & justify each problem with heuristics
How to do heuristic evaluation

Tips for doing a good heuristic evaluation:
- Ground your evaluation in known usability guidelines
  - Justify each problem you list by appealing to a heuristic
  - Explain how the heuristic is violated
  - Remove subjectivity involved in inspections:
    - Can’t just say “That’s an ugly yellow colour”
How to do heuristic evaluation (cont’d)

- List every problem you find
  - even if one interface element has several problems
- Go through the interface at least twice
  - Once to get the feel of the system
  - Again to focus on particular interface elements
- Don’t limit yourself to the one set of heuristics (Schneiderman, Norman, Nielsen) + Fitz’s law, perceptual fusion etc.
Nielsen’s heuristics (1)

Meet expectations

1. Match the real world
2. Consistency & standards
3. Help & documentation
Nielsen’s heuristics (2)

User is boss

- 4. User control & freedom
- 5. Visibility of system status
- 6. Flexibility & efficiency
Nielsen’s heuristics (3)

**Errors**
- 7. Error prevention
- 8. Recognition, not recall
- 9. Error reporting, diagnosis, and recovery
Nielsen’s heuristics (4)

Keep it simple

- 10. Aesthetic & minimalist design
Welcome, Ben Bitdiddle.

You have 2 item(s) in your shopping cart.
To remove an item, check "Remove" box & click "Recalculate".
Shipping Calculator below.

*There is a problem with your order.*

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
<th>Quantity</th>
<th>UnitPrice</th>
<th>ExtPrice</th>
</tr>
</thead>
<tbody>
<tr>
<td>323022</td>
<td>Pinnaclce Clean Plus Version 4.0 Retail *** (Free 2nd Day) ***</td>
<td>1</td>
<td>$61.00</td>
<td>$61.00</td>
</tr>
<tr>
<td></td>
<td>Remove</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80098-21</td>
<td>Corsair VS1GBKIT400 1GB Kit DDR400 PC3200 Value Select Memory Retail (out of stock)</td>
<td>1</td>
<td>$179.00</td>
<td>$179.00</td>
</tr>
</tbody>
</table>

For more information about tax, please click here.

Shipping Promotion details. Please read

*Note: Discount will be applied during check out*

<table>
<thead>
<tr>
<th>Coupon Code:</th>
<th>Ship to Zip Code:</th>
<th>Apply</th>
<th>Calculate Shipping Charge</th>
</tr>
</thead>
</table>

Have not made up your mind? Save all the items in your shopping cart!

Cart Title: | Save Shopping Cart |

Return to old shopping cart:

Cart Name: | Load Shopping Cart |
Heuristics evaluation is not user testing

- Evaluator is not the user either
- Maybe closer to being a typical user than you are
- Analogy: code inspection vs. testing
- HE finds problems that UT often misses
  - Inconsistent fonts
  - Fitz’s Law problems
Hints for better heuristic evaluation

- Use multiple evaluators
  - Different evaluators find different problems
  - The more the better, but diminishing returns
  - Nielsen recommends 3-5 evaluators

- Alternate heuristic evaluation with user testing around the iterative design cycle
  - Each method finds different problems
  - Heuristic evaluation is cheaper

- It is OK for observer to help evaluator as long as the problem has already been noted

- This wouldn’t be OK in a user test
Why multiple evaluators

Heuristic evaluation of a banking system;
19 evaluators;
16 usability problems;
Proportion of usability problems in an interface found by heuristic evaluation using various numbers of evaluators

Source: Nielsen
Curve showing how many times the benefits are greater than the costs for heuristic evaluation of a sample project. Optimal number of evaluators = 4, with benefits that are 62 times greater than the costs.

Source: Nielsen
Formal evaluation process

1. **Training**
   - Meeting for design team & evaluators
   - Introduce application
   - Explain user population, domain, scenarios

2. **Evaluation**
   - Evaluators work separately
   - Generate written report, or oral comments recorded by an observer
   - Focus on generating problems, not on ranking their severity yet
   - 1-2 hours per evaluator
Formal evaluation process (cont’d)

3. Severity Rating
- Evaluators prioritize all problems found (not just their own)
- Take the mean of the evaluators ratings

4. Debriefing
- Evaluators & design team discuss results, brainstorm solutions
Severity ratings

- Used to allocate the most resources to the most serious problems
- Rough estimate of the need for additional usability efforts

- 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 problem that users can overcome once they know about it or will users repeatedly be bothered by the problem?
Severity scale

0  = I don't agree that this is a usability problem at all
1  = Cosmetic problem only: need not be fixed unless extra time is available on project
2  = Minor usability problem: fixing this should be given low priority
3  = Major usability problem: important to fix, so should be given high priority
4  = Usability catastrophe: imperative to fix this before product can be released
Evaluating prototypes

- Heuristic evaluation works on:
  - Sketches
  - Paper prototypes
  - Buggy implementations

- Missing-element problems are harder to find on sketches
- Because you’re not actually using the interface, you are not blocked by features absence
- Look harder for missing elements
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)
Cognitive walkthrough

- A formalized way of imagining people’s thoughts and actions when they use an interface for the first time.

- Who should do the walkthrough and when:
  - If designing a small piece of the interface, you (the designer) can do informal walkthroughs to monitor the progress of the design.
  - In later design iterations, walkthrough with a group of designers (and possibly users) for a complete task.

A tool mainly for developing the interface.
What you need

- The prototype to be evaluated
- A task description for the task that the UI is intended to support
- A complete, sequential list of actions that implement the given task
- An idea of who the users will be and what kind of experience they will bring to the job
How to do the walkthrough

- Try to tell a story about what the user would select each action in the list of correct actions
- Critique the story by using four main questions
How to do the walkthrough (cont’d)

1. Will users be trying to produce whatever effect the action has?
2. Is the desired action visible at the interface? Will users see the control (button, menu etc.) for the action? Will users be able to select the desired action from others currently visible?
3. Once users find the control, will they recognize that it produces the effect they want?
4. Will users understand the feedback after the action is performed?
1. Will users be trying to produce whatever effect the action has?

- Example: a portable computer with a slow-speed mode for its processor for saving battery power.
- First action for computationally expensive tasks: toggle the processor into a high-speed mode
- Will novice users try to do this?
Question 2. Is the desired action visible at the interface?

- Often a problem, because often important controls are hidden.
- Example: office copier with many of its buttons hidden under a small door
- Task: make double sided-copies
- The help sheet tells users which button to choose
- Problem: how to find the button
Question 3.

Once users find the control, will they recognize that it produces the effect they want?

- Issues to be considered:
  - unambiguous label of the control
  - Intuitive icon
Question 4.

- After the action is taken, will users understand the feedback they get?
  - deals with bridging the goal of evaluation
  - it is important to record not only the user’s actions but also the system’s responses
  - Consistent with heuristic evaluation guidelines
Example of walkthrough: Video remote control

- Task: record a TV program using a remote control
- (A): initial display ; (B): display after the record button has been pressed
Example of walkthrough (cont’d)

- Task: Program the video to record a program starting at 19:00, finishing at 20:15 on channel 4 on July 1st

- Identify the action sequence in terms of:
  - User Action: A, B, C, ...
  - System Response: A, B, C, ...

Action – response I

- Act. A: press the “timed record” button
- Resp. A: display moves to timer mode. Flashing cursor appears after “start”
- Act. B: press digits 19:00
- Resp. B: each digit is displayed as typed, flashing cursor moves to next position
- Act. C: press the “timed record” button
- Resp. C: flashing cursor moves to “End”
- Resp. D: each digit is displayed as typed and flashing cursor moves to next position
Action – response II

- Act. E: press the “timed record” button
- Resp. E: flashing cursor moves to “channel”
- Act. F: press digit 3
- Resp. F: digit is displayed as typed and flashing cursor moves to next position
- Act. G: press the “timed record” button
- Resp. G: flashing cursor moves to “Date”
- Resp. H: each digit is displayed as typed and flashing cursor moves to next position
Action – response III

- Act. I: press the “timed record” button
- Resp. I: stream number in top right corner of display flashes

- Act. J: press the “transmit” button
- Resp. J: details are transmitted to video player and display returns to normal mode
Walkthrough example

- For each action we must record the answers to the 4 questions about the usability of the system.
- We’ll analyze action A.
Studying action A

- Act. A: press the “timed record” button

Question 1: will the users try to produce whatever effect the actions has?

The interface provides no indication that the user needs to press the “timed record” button.
Still studying action A

Question 2: will users be able to notice that the correct action is available?
Answer...

Question 3: once users find the correct action at the interface, will they know that it is the right one for the effect they are trying to produce?
Answer...
Final study of action A

- Question 4: after the action is taken, will users understand the feedback they get?
  
  Answer:
Usability problem

- potential usability problem stems from the first action: how to find the correct icon to start the TV program recording

- The walkthrough proceeds to the analysis of the other actions, always trying to answer the four basic questions.
Cognitive walkthroughs and usability measures

- Good for designing in:
  - Visibility
  - System feedback
  - Dialogue design
  - Information noise

- Poor for designing in:
  - Consistency
  - Flexibility
  - Efficiency
  - Satisfaction