Questions?
Evaluating Without Users

- **Cognitive Walkthroughs**
  - Task oriented
  - Path through interface pre-determined
  - One analyzer

- **Action Analysis**
  - KLM

- **Heuristic Analysis**
  - Interface (not task) oriented
  - Overall examination. Path through interface NOT pre-determined
  - Several analyzers
Cognitive Walkthrough

Requirements:
• Description or prototype of interface
• Task description
• List of actions to complete task
• User background

What you look for:
• Will users know to perform the action?
• Will users see the control?
• Will users know the control does what they want?
• Will uses understand the feedback?
Heuristic Analysis

• “Rules of thumb” that describe features of usable systems
  – Can be used as design principles
  – Can be used to evaluate a design

• Pros and cons
  – Easy and inexpensive
    • Performed by expert
    • No users required
    • Catches many design flaws
  – More difficult than it seems
    • Not a simple checklist
    • Cannot assess how well the interface will address user goals
Usability Engineering

- Introduced by Nielsen (1994)
- Can be performed on working UI or sketches
- Requires a small set of evaluators to examine the UI
  - Check compliance with usability principles
    - *Each evaluator works independently*
    - *Go through the interface several times with different perspectives*
  - All reviews are aggregated in one final usability report
Nielsen's evaluation phases (1-2)

• Pre-evaluation training
  – Provide the evaluator with domain knowledge if needed

• Evaluation
  – First step: get a feel for flow and scope
  – Second step: focus on specific elements
    • Multiple passes is better
    • Create a list of all problems
    • Rate severity of problem
Nielsen's evaluation phases (3-4)

- Severity rating
  - Performed by individuals
  - Then aggregated by group
  - Establishes a ranking between problem
  - Reflects frequency, impact and persistence
    - Cosmetic, minor, major and catastrophic

- Debriefing
  - Discuss outcome with design team
  - Suggest potential solutions
  - Assess how hard things are to fix
### Evaluation Aggregation

#### Machine 6 - Zoomable

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Severity</td>
<td>Average Severity</td>
<td>Count</td>
</tr>
<tr>
<td>2</td>
<td>(1-Low =&gt; 5-High)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4.5</td>
<td>4.0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>3.0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>?</td>
<td>3.0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>3.0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>3.0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1.4</td>
<td>2.5</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>2.0</td>
<td>1</td>
</tr>
</tbody>
</table>
Nielsen’s heuristics

- Simple and natural dialog
- Speak the users’ language
- Minimize user memory load
- Consistency
- Feedback
- Clearly marked exits
- Shortcuts
- Prevent errors
- Good error messages
- Provide help and documentation
Simple and natural dialog

From Cooper’s “The inmates are running the asylum”
Simple and natural dialog

• Present information in natural order

From Cooper’s “About face 2.0”

• Simple is good
  – Remove or hide irrelevant or rarely needed information
    • They compete with important information on screen
      – Pro: Palm Pilot
      – Against: Dynamic menus
  – Use windows frugally
    • Avoid complex window management
Speak the users’ language

• Use a language compatible with users’ conceptual model
  – Example: withdrawing money at an ATM

• Use meaningful mnemonics, icons and abbreviations
Minimize user memory load

- Promote recognition over recall
  - Recognition is easier than recall

- Describe expected input clearly
  - Don’t allow for incorrect input

- Create orthogonal command systems
  - Using generic commands that can be applied to all interface objects
Consistency

• Be consistent in
  – Command design
    • Same action, same effect in equivalent situations
  – Graphic design
    • Input format
    • Output format
  – Flow design
    • Similar tasks are handled in similar ways

• Consistency promotes skills acquisition and/or transfer
Feedback (Semantic)

• Users should always be aware of what is going on
  – So that they can make informed decision
    • *Be specific*

  – But do not overburden users!

  – Provide redundant information

Feedback: Toolbar, cursor, ink
Feedback (Time)

- Different feedback time scales
  - Shall I wait for that task to finish or go for coffee?
    - > 10s  User will switch to another task while waiting
    - 10s  Difficult to stay focused
    - 1s  Delay but user’s flow of thought is uninterrupted
    - .1s  Causality

- Different techniques
  - Short transaction: hour glass cursor
  - Longer transaction: estimate of time left
    - An overestimate is always better!
Clearly marked exits

• Users don’t like to be trapped!

• Strategies
  – Cancel button (or Esc key) for dialog
    • *Make the cancel button responsive!*
  – Universal undo
Shortcuts (I)

• Expert users should be able to perform operations rapidly
  – Try to limit the training necessary to access advanced features

• Strategies
  – Keyboard and mouse accelerators
    • menu shortcuts and function keys
    • command completion, command abbreviations and type-ahead
  – Toolbars and tool palettes
    • Trade screen real estate for rapid access
  – Navigation jumps
    • History systems
      – 60% pages are revisits
Shortcuts (II)

Shortcuts (I)

- Expert users should be able to perform operations rapidly
  - Try to limit the training necessary to access advanced features

- Strategies
  - Keyboard and mouse accelerators
    - menu shortcuts and function keys
    - command completion, command abbreviations and type-ahead
  - Toolbars and tool palettes
    - Trade screen real estate for rapid access
  - Navigation jumps
    - History systems
      - 60% pages are revisits

Marking menu and the like but still a research issue
Preventing errors

• Error types
  – Mistakes
    • Conscious decision with unforeseen consequences
  – Slips
    • Automatic behaviors kicking in
      – Drive to the store, end-up in the office
      – Press enter one time too many…
    • Mode errors
      – Forget the mode the application is in
    • Loss of activation
      – Forget what your goals were
Designing for slips

An ounce of prevention is worth more than a pound of cure!

• Examples
  – Design modeless interfaces
  – Instead of confirmations provide undo mechanisms
  – Check for reasonable input
    • Be prepared to handle several formats
    • Make entering a incorrect format impossible
  – Make the current goal clear
    • Prevent lost of activations
Forcing functions

• Interlock mechanisms
  – Require step A before step B can be performed
  – Ex: Switching from P to D in a car requires pressing brake pedal

• Lockin mechanisms
  – Process continues unless user removes constraint before stopping it
  – Ex: No eject button for floppy disk on Mac

• Lockout mechanisms
  – Process won’t occur unless user removes constraint before starting it
  – Ex: Basement stairway
Questions
Dealing with errors

• People will make errors!
  – You can ignore them
    • Generally very confusing
  – You can correct them automatically
    • Spelling corrector
    • But is the system right 100% of the time?
  – You can discuss it
    • But novice/expert tradeoff
  – You can try to teach the user what to do
    • Office assistant

• Respect users feelings!
  – The user is never wrong
Good error messages

Please take note:

The file you are editing, "Prose.txt", which resides on volume "NetOne", is unavailable due to an unidentified network failure. No data has been lost, however.

Scope: The file will be automatically restored when the network connection is re-established.

Action: A local copy of "Prose.txt" will be saved to the volume titled "Drive C:" in folder "Local Save" on your Desktop. If you would like to save to a different location, press Save As... below; otherwise, just press OK.

More: This program does not have the ability to diagnose the problem further. Please contact your network administrator for further information.

From Cooper’s “About Face 2.0”
Good error messages

• Provide meaningful error messages
  – Explain the problem in terms of the user conceptual model
  – Don’t make the user feel stupid
  – Offer a way to correct the problem

  – Compare
    • *Error 25: access denied*
    • *Cannot open “chapter 5” because “Microsoft Word” is not installed. Do you want to use Notepad instead?*
Provide help and documentation

• Providing help is not an excuse for poor design!
  – Saving a couple of line of code or writing several pages of documentation?
  – Users don’t like to read manuals
    • They prefer to learn while making progress toward their goals

• Most users will stay at the intermediate level
  – Need reminders and a clear learning path
  – Need a quick way to access critical information
    • Online documentation and good search tool
Types of help (I)

- Tutorial and/or getting started manuals
  - Presents the system conceptual model
    - *Basis for successful explorations*
  - Provides on-line tours and demos
    - *Demonstrates basic features*

- Reference manuals
  - Designed with experts in mind

- Reminders
  - Short reference cards, keyboard templates, tooltips…

- “Show me” videos
Types of help (II)

• Wizards
  – Walks user through typical tasks
    • Users feel they are losing control
    • What if I do not have the information requested?

• Tips
  – Migration path to learning new features
  – Can become boring and tedious
Types of help (II)

• Context sensitive help
Shneiderman’s “Golden Rules”

• Strive for consistency
• Cater to universal usability
• Offer informative feedback
• Design dialogs to yield closure
• Prevent errors
• Permit easy reversal of actions
• Support internal locus of control
• Reduce short-term memory load
Flow

• Challenge and require skill
• Concentrate and avoid interruption
• Maintain control
• Speed and feedback
• Transformation of time
  – What is RSD?
Example: NoteLens

Welcome to NoteLens!

You are on your way to managing all of your notes, thoughts and ideas. To get started:

1. Create new notes through the "File→New Note" menu (or more simply, Ctrl-N). There is no need to give notes names or locations, and you never have to explicitly save them.
2. You can see that the first line of each note shows up in a list on the top part of NoteLens. You can read or modify a note by clicking on it in this list.
3. Find notes by typing any words (or parts of words) in the "Find Box" at the top of NoteLens.
4. Uncheck the "Search Subject Only" box to make NoteLens search all of the text in your notes.

To learn more, select the "Help→NoteLens Help" menu. You can delete this note by selecting it in the Find Box and pressing the Delete key.
Example: EverNote
Keep track of recipes

Examples of what to put in this section
- Recipes and menus you find on the Web
- Recipes you type into Microsoft Office OneNote 2007
- Links to online cooking discussion groups

Tips
- Create a new page for each recipe. As your collection grows over time, create a separate notebook with specific sections to organize all of your recipes.
- Copy information from a Web site into OneNote 2007 as text. If you use Windows Internet Explorer, look for the OneNote button in the toolbar. Select the information you want to copy and click this button to send the information to OneNote 2007 in the Unfiled Notes section.
- When you copy information from a Web browser and paste onto a page, a reference link is inserted on the page so you can always return to your source.
- When you need to find a recipe OneNote 2007, use the search feature (CTRL+F). OneNote 2007 searches text, text within images, ink writing, and audio recordings.
- Want to share a recipe with someone else? Go to the page with the recipe you want to share. On the File menu, click E-mail or press CTRL+SHIFT+E to send recipes to other people. They will be able to read the e-mail message you send even if they don’t have OneNote 2007.
Exercise: Heuristic Evaluation

Find fastest way to get from campus to Silver Spring metro station along with cost.

www.wmata.com