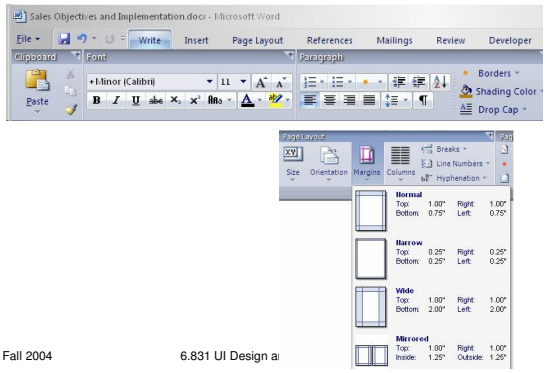
Lecture 14:
Heuristic Evaluation

UI Hall of Fame or Shame?



Nielsen's 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

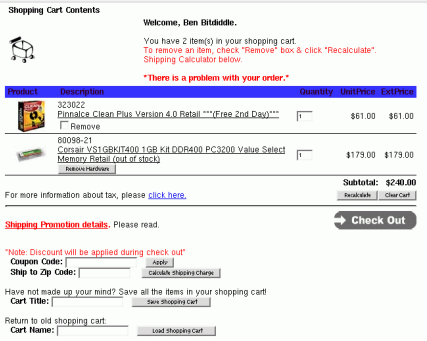
Heuristic Evaluation

- Performed by an expert
- Steps
  - Inspect UI thoroughly
  - Compare UI against heuristics
  - List usability problems
    - Explain & justify each problem with heuristics

How To Do Heuristic Evaluation

- Justify every problem with a heuristic
  - "Too many choices on the home page (Aesthetic & Minimalist Design)"
  - Can't just say "I don't like the colors"
- List every problem
  - Even if an interface element has multiple 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 10 heuristics
  - We've seen others: affordances, visibility, Fitts's Law, perceptual fusion, color principles
  - But the 10 heuristics are easier to compare against

Example



Heuristic Evaluation Is Not User Testing

- Evaluator is not the user either
  - Maybe closer to being a typical user than you are, though
- Analogy: code inspection vs. testing
- HE finds problems that UT often misses
  - Inconsistent fonts
  - Fitts's Law problems
- But UT is the gold standard for usability

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
  - Each method finds different problems
  - Heuristic evaluation is cheaper
- It's OK for observer to help evaluator
  - As long as the problem has already been noted
  - This wouldn't be OK in a user test

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
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

- Contributing factors
  - Frequency: how common?
  - Impact: how hard to overcome?
  - Persistence: how often to overcome?
- Severity scale
  1. Cosmetic: need not be fixed
  2. Minor: needs fixing but low priority
  3. Major: needs fixing and high priority
  4. Catastrophic: imperative to fix

Evaluating Prototypes

- Heuristic evaluation works on:
  - Sketches
  - Paper prototypes
  - Unstable prototypes
- "Missing-element" problems are harder to find on sketches
  - Because you're not actually using the interface, you aren't blocked by feature's absence
  - Look harder for them

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)"

Suggested Report Format

- What to include:
  - Problem
  - Heuristic
  - Description
  - Severity
  - Recommendation (if any)
  - Screenshot (if helpful)

12. 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