CS 160: Evaluation

Professor John Canny
Spring 2006

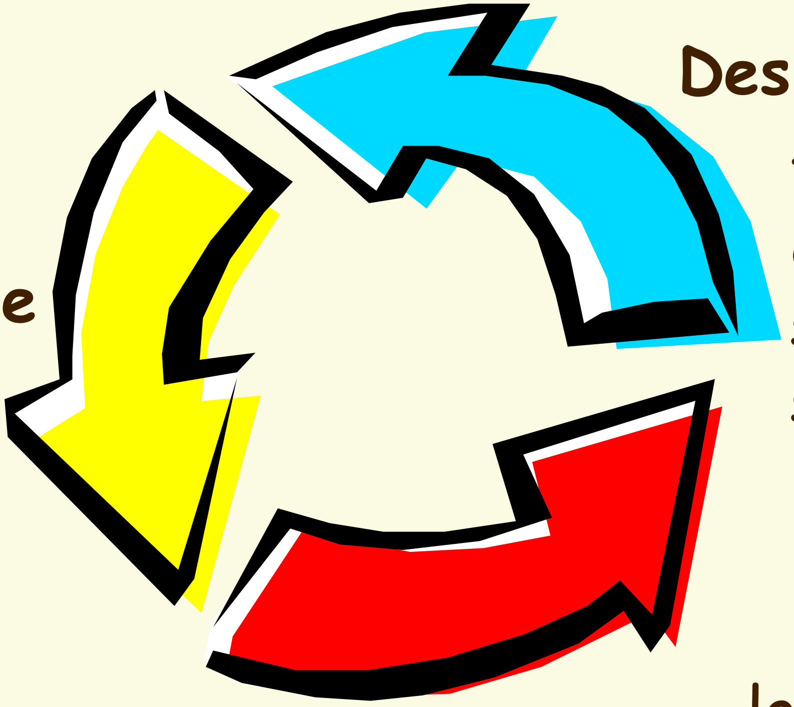
Outline

- User testing process
- Severity and Cost ratings
- Discount usability methods
- Heuristic evaluation
- HE vs. user testing

Outline

- User testing process
- Severity and Cost ratings
- Discount usability methods
- Heuristic evaluation
- HE vs. user testing

Iterative Design



Design

task analysis
contextual inquiry
scenarios
sketching

Prototype

low-fi
paper, DENIM

Evaluate

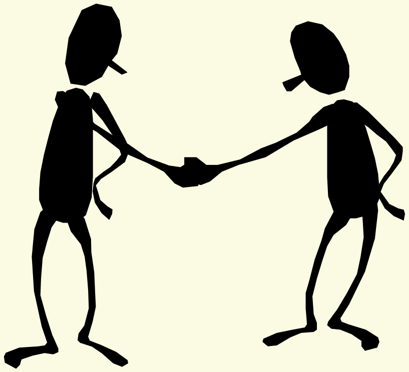
low-fi testing,
Heuristic eval

Preparing for a User Test

- Objective: narrow or broad?
- Design the tasks
- Decide on whether to use video/audio
- Choose the setting
- Representative users

User Test

- Roles:
  - Greeter
  - Facilitator: Help users to think aloud…
  - Observers: record “critical incidents”



Critical Incidents

- Critical incidents are unusual or interesting events during the study.
- Most of them are usability problems.
- They may also be moments when the user:
  - got stuck, or
  - suddenly understood something
  - said “that’s cool” etc.

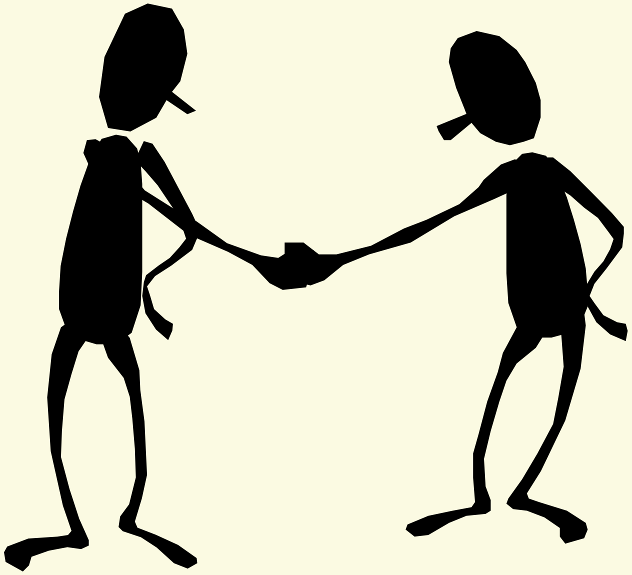
The User Test

- The actual user test will look something like this:
  - Greet the user
  - Explain the test
  - Get user's signed consent
  - Demo the system
  - Run the test (maybe ½ hour)
  - Debrief

10 steps to better evaluation

1. Introduce yourself

   some background will help relax the subject.



10 steps

2. Describe the purpose of the observation (in general terms), and set the participant at ease
   - You're helping us by trying out this product in its early stages.
   - If you have trouble with some of the tasks, it's the product's fault, not yours. Don't feel bad; that's exactly what we're looking for.

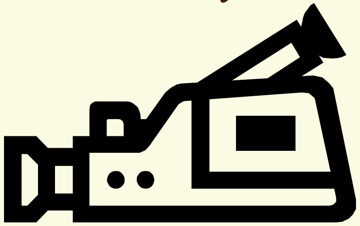
10 steps (contd.)

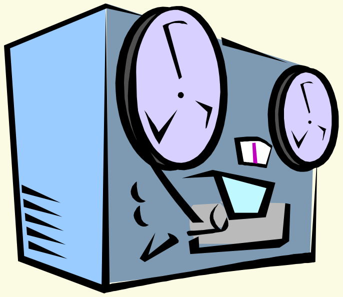
3. Tell the participant that it's okay to quit at any time, e.g.:
    - Although I don't know of any reason for this to happen, if you should become uncomfortable or find this test objectionable in any way, you are free to quit at any time.

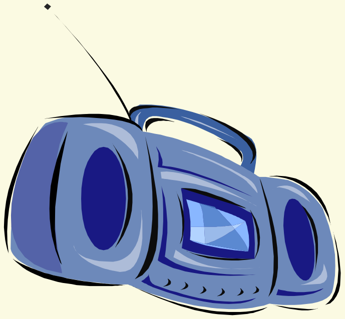
10 steps (contd.)

4. Talk about the equipment in the room.

- Explain the purpose of each piece of equipment (hardware, software, video camera, microphones, etc.) and how it is used in the test.

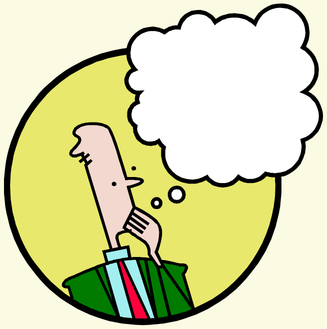






10 steps (contd.)

5. Explain how to “think aloud.”
   - Explain why you want participants to think aloud, and demonstrate how to do it. E.g.:
   - We have found that we get a great deal of information from these informal tests if we ask people to think aloud. Would you like me to demonstrate?



10 steps (contd.)

6. Explain that you cannot provide help.

10 steps (contd.)

7. Describe the tasks and introduce the product.

   - Explain what the participant should do and in what order. Give the participant written instructions for the tasks.
   - Don't demonstrate what you're trying to test.

10 steps (contd.)

8. Ask if there are any questions before you start; then begin the observation.

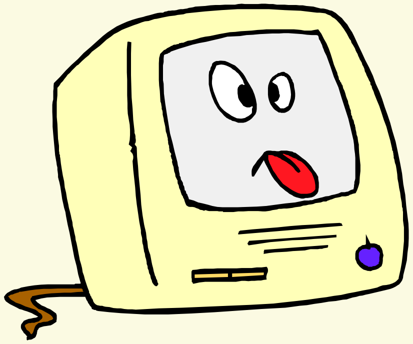
10 steps (contd.)

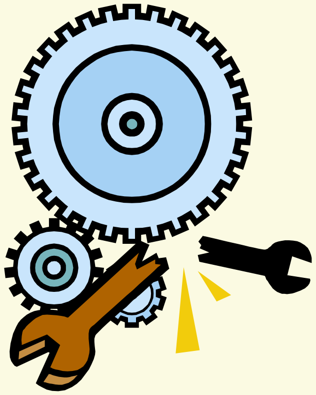
9. Conclude the observation. When the test is over:
   - Explain what you were trying to find.
   - Answer any remaining questions.
   - discuss any interesting behaviors you would like the participant to explain.

10 steps (contd.)

10. Use the results.

- When you see participants making mistakes, you should attribute the difficulties to faulty product design, not to the participant.





Using the Results

- Update task analysis and rethink design
  - Rate severity & ease of fixing problems
  - Fix both severe problems & make the easy fixes
- Will thinking aloud give the right answers?
  - Not always
  - If you ask a question, people will always give an answer, even it is has nothing to do with the facts
  - Try to avoid leading questions

Questions?

High-order summary:

- Follow a loose master-apprentice model
- Observe, but help the user describe what they're doing
- Keep the user at ease

Outline

- User testing process
- Severity and Cost ratings
- Discount usability methods
- Heuristic evaluation
- HE vs. user testing

Outline

- User testing process
- Severity and Cost ratings
- Discount usability methods
- Heuristic evaluation
- HE vs. user testing

Severity Rating

- Used to allocate resources to fix problems
- Estimate of consequences of that bug
- Combination of
  - Frequency
  - Impact
  - Persistence (one time or repeating)
- Should be calculated after all evaluations are in
- Should be done independently by all judges

Severity Ratings (cont.)

0 - don't agree that this is a usability problem

1 - cosmetic problem

2 - minor usability problem

3 - major usability problem; important to fix

4 - usability catastrophe; imperative to fix

Cost (to repair) ratings

Later in the development process, it will be important to rate the cost (programmer time) of fixing usability problems.

A similar rating system is usually used, but the ratings are made by programmers rather than usability experts or designers.

With both sets of ratings, the team can optimize the benefit of programmer effort.

Debriefing

- Conduct with evaluators, observers, and development team members.
- Discuss general characteristics of UI.
- Suggest potential improvements to address major usability problems.
- Make it a brainstorming session
  - little criticism until end of session

Break

Note: midterm coming up on Monday 2/27

Outline

- User testing process
- Severity and Cost ratings
- Discount usability methods
- Heuristic evaluation
- HE vs. user testing

Outline

- User testing process
- Severity and Cost ratings
- Discount usability methods
- Heuristic evaluation
- HE vs. user testing

Discount Usability Engineering

- Cheap
  - No special labs or equipment needed
  - The more careful you are, the better it gets
- Fast
  - On order of 1 day to apply
  - Standard usability testing may take a week
- Easy to use
  - Can be taught in 2-4 hours

Cost of user testing

- Its very expensive – you need to schedule (and normally pay) many subjects.
- It takes many hours of the evaluation team's time.
- A user test can easily cost $10k's

Discount Usability Engineering

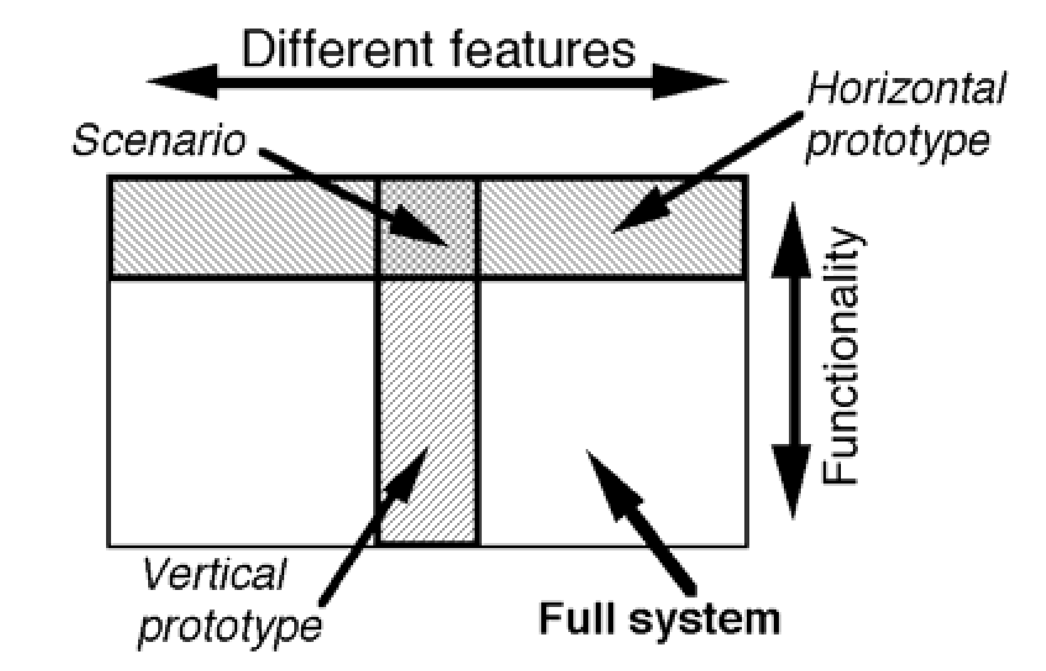
- Based on:
  - Scenarios
  - Simplified thinking aloud
  - Heuristic Evaluation
  - Some other methods…

Scenarios

- Run through a particular task execution on a particular interface design
- Build just enough of the interface to support that
- A scenario is a simplest possible prototype

Scenarios

- Eliminate parts of the system
- Compromise between horizontal and vertical prototypes



Simplified thinking aloud

- Bring in users
- Give them real tasks on the system
- Ask them to think aloud as in other methods
- No video-taping – rely on notes
- Less careful analysis and fewer testers

Other budget methods

- Walkthroughs
  - Put yourself in the shoes of a user
  - Like a code walkthrough
- Action analysis
  - GOMS (later…)
- On-line, remote usability tests
- ***Heuristic evaluation***

Outline

- User testing process
- Severity and Cost ratings
- Discount usability methods
- Heuristic evaluation
- HE vs. user testing

Outline

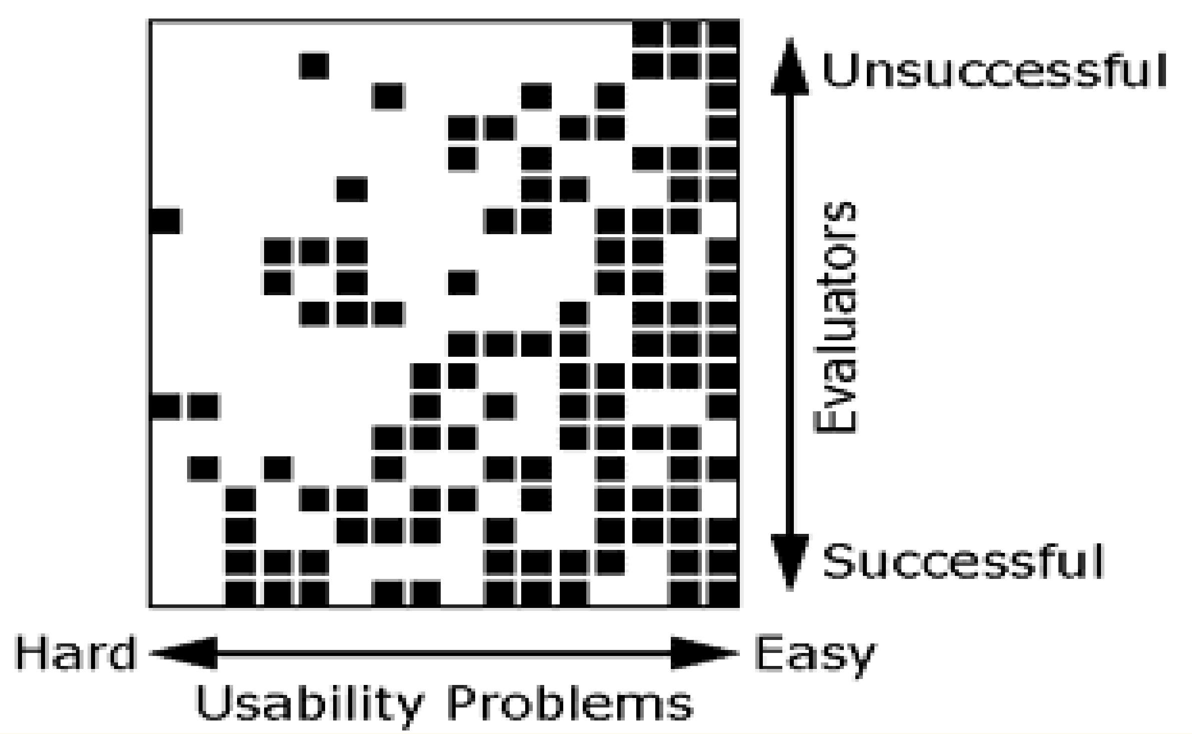
- User testing process
- Severity and Cost ratings
- Discount usability methods
- Heuristic evaluation
- HE vs. user testing

Heuristic Evaluation

- Developed by Jakob Nielsen
- Helps find usability problems in a UI design
- Small set (3-5) of evaluators examine UI
  - Independently check for compliance with usability principles ("heuristics")
  - Different evaluators will find different problems
  - Findings are aggregated afterwards
- Can be done on a working UI or on sketches

Why Multiple Evaluators?

- Every evaluator doesn't find every problem
- Good evaluators find both easy & hard ones



Heuristic Evaluation Process

- Evaluators go through UI several times
  - Inspect various dialogue elements
  - Compare with list of heuristics
- Heuristics
  - Nielsen's "heuristics"
  - Supplementary list of category-specific heuristics
  - Get them by grouping usability problems from previous user tests on similar products
- Use violations to redesign/fix problems

Heuristics (original)

- **H1-1:** Simple & natural dialog
- **H1-2:** Speak the users' language
- **H1-3:** Minimize users' memory load
- **H1-4:** Consistency
- **H1-5:** Feedback
- **H1-6:** Clearly marked exits
- **H1-7:** Shortcuts
- **H1-8:** Precise & constructive error messages
- **H1-9:** Prevent errors
- **H1-10:** Help and documentation

Revised Heuristics

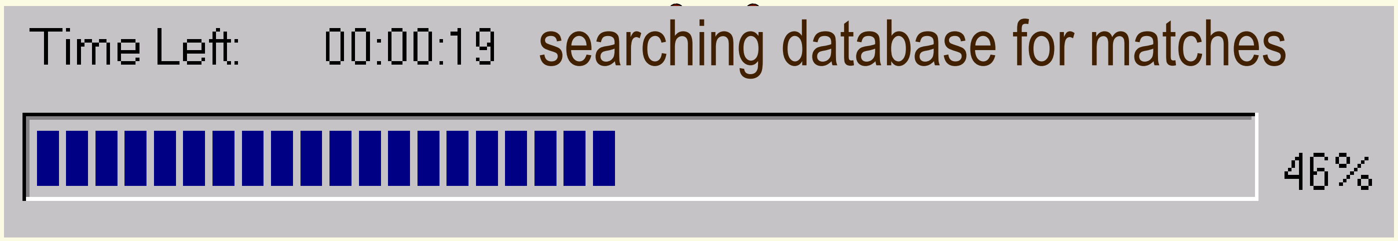
Also developed by Nielsen.

- Based on factor analysis of 249 usability problems
- A prioritized, independent set of heuristics

Revised Heuristics

- **H2-1**: visibility of system status
- **H2-2**: Match system and real world
- **H2-3**: User control and freedom
- **H2-4**: Consistency and standards
- **H2-5**: Error prevention
- **H2-6**: Recognition rather than recall
- **H2-7**: Flexibility and efficiency of use
- **H2-8**: Aesthetic and minimalist design
- **H2-9**: Help users recognize, diagnose and recover from errors
- **H2-10**: Help and documentation

Heuristics (revised set)

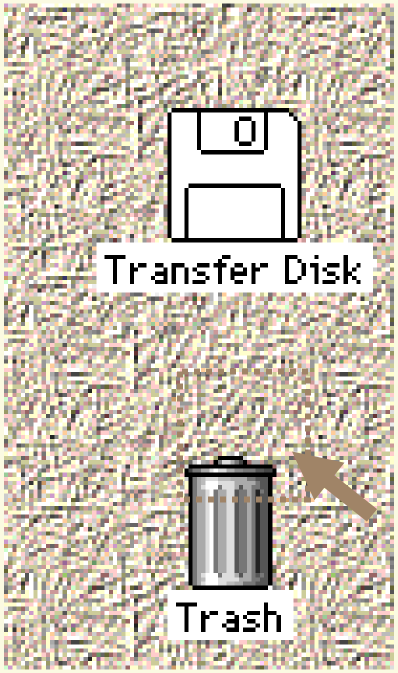


Time Left: 00:00:19 searching database for matches

46%

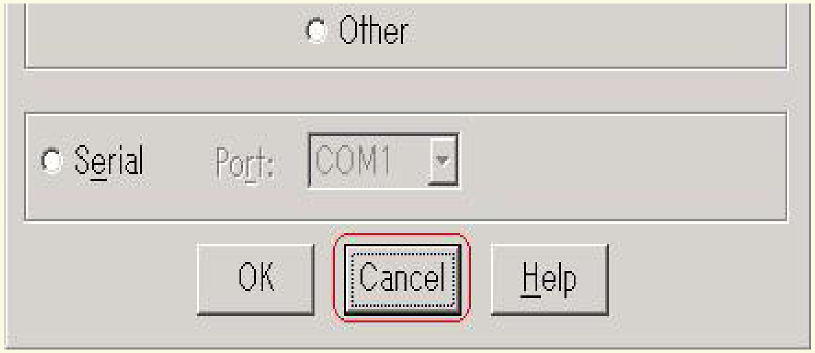
- H2-1: Visibility of system status
  - keep users informed about what is going on
  - 0.1 sec: no special indicators needed
  - 1.0 sec: user tends to lose track of data
  - 10 sec: max. duration if user to stay focused on action
  - for longer delays, use percent-done progress bars

Heuristics (cont.)



- H2-2: Match between system & real world
  - speak the users' language
  - follow real world conventions
- Bad example: Mac desktop
  - Dragging disk to trash
  - should delete it, **not** eject it

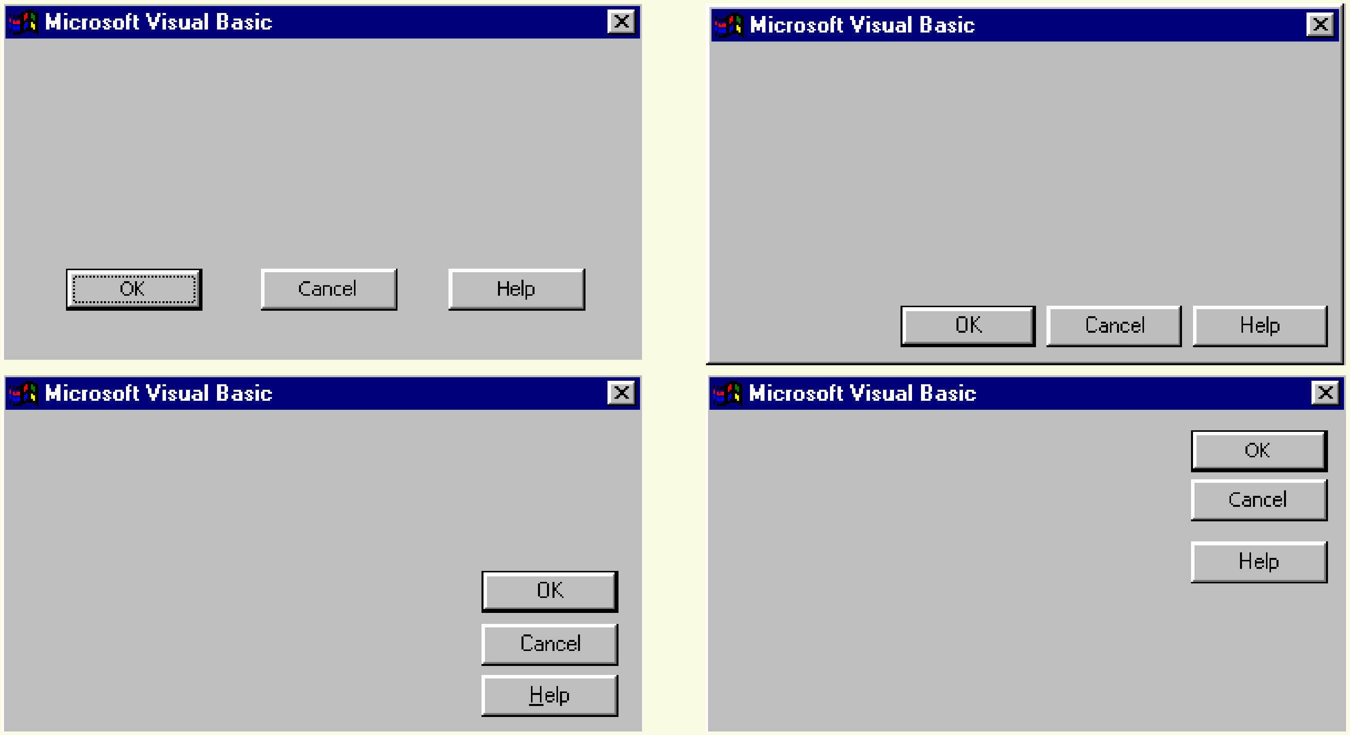
Heuristics (cont.)



H2-3: User control & freedom

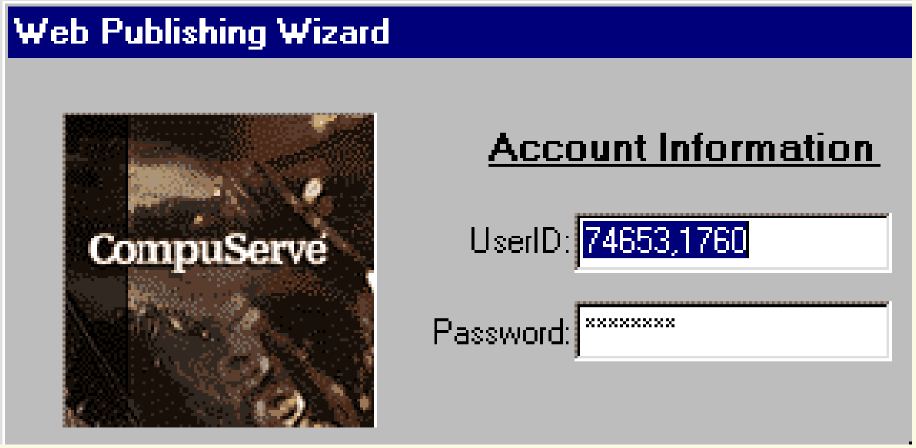
- "exits" for mistaken choices, undo, redo
- don't force down fixed paths like BART ticket machine…

Heuristics (cont.)



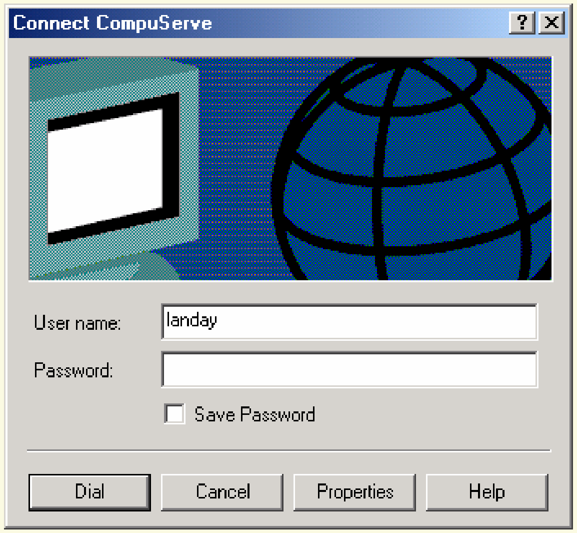
- H2-4: Consistency & standards

Heuristics (cont.)



UserID: 74653,1760

Password: *********



User name: landay

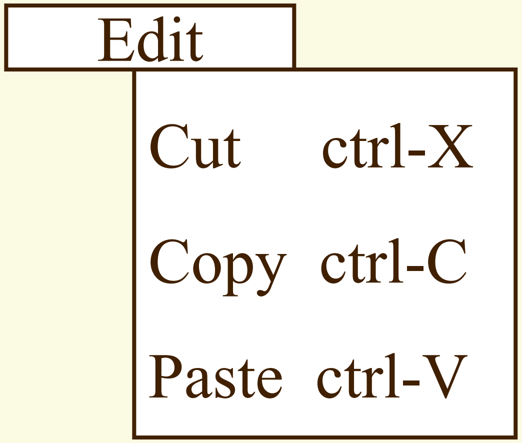
Password:

□ Save Password

- H2-5: Error prevention
- H2-6: Recognition rather than recall
- make objects, actions, options, & directions visible or easily retrievable

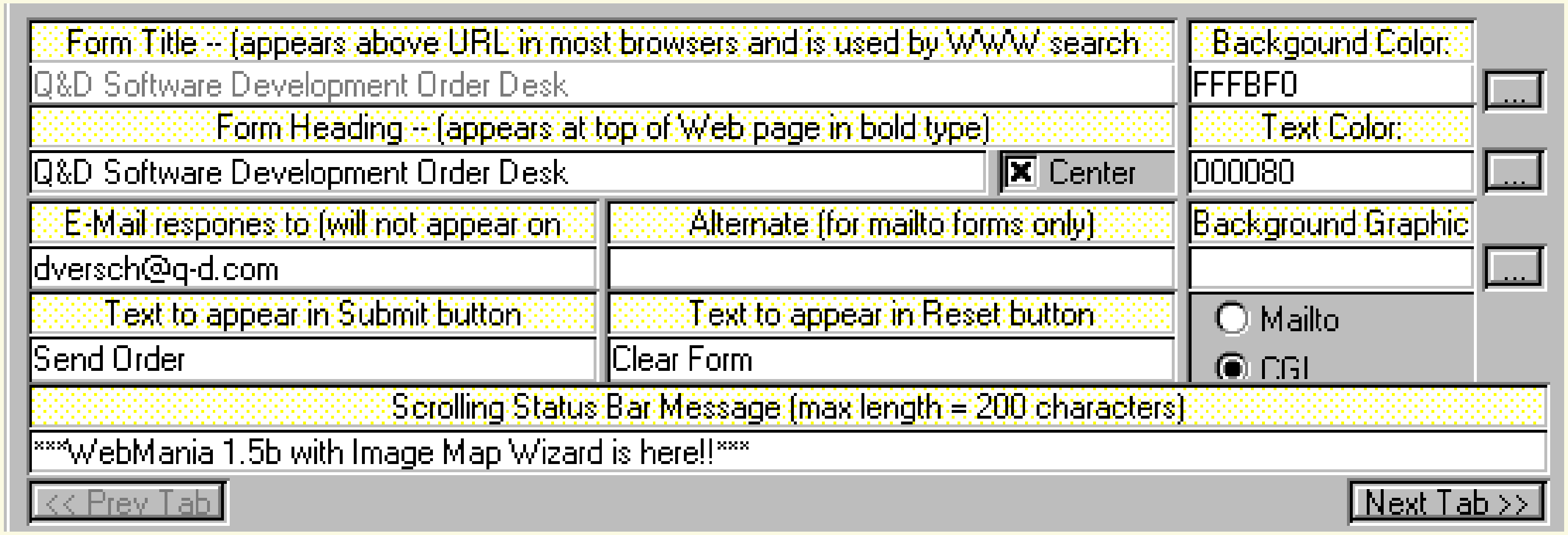
- MS Web Pub. Wiz.
- Before dialing, asks for id & password
- When connecting, asks again for id & pw

Heuristics (cont.)



- H2-7: Flexibility and efficiency of use
  - accelerators for experts (e.g., gestures, keyboard shortcuts)
  - allow users to tailor frequent actions (e.g., macros)

Heuristics (cont.)



Form Title -- (appears above URL in most browsers and is used by WWW search

Q&D Software Development Order Desk

Background Color:

FFFBF0

Form Heading -- (appears at top of Web page in bold type)

Q&D Software Development Order Desk

☒ Center

Text Color:

000080

E-Mail respones to (will not appear on

dversch@q-d.com

Alternate (for mailto forms only)

Background Graphic

Text to appear in Submit button

Send Order

Text to appear in Reset button

Clear Form

○ Mailto

● CGI

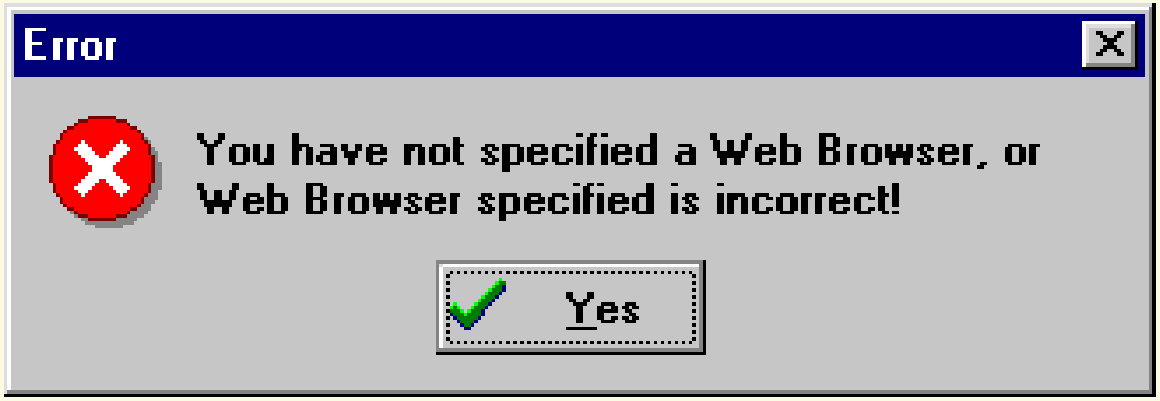
Scrolling Status Bar Message (max length = 200 characters)

WebMania 1.5b with Image Map Wizard is here!!

H2-8: Aesthetic and minimalist design

- no irrelevant information in dialogues

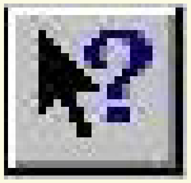
Heuristics (cont.)



You have not specified a Web Browser, or Web Browser specified is incorrect!

- H2-9: Help users recognize, diagnose, and recover from errors
  - Error messages in plain language
  - Precisely indicate the problem
  - Constructively suggest a solution

Heuristics (cont.)



- H2-10: Help and documentation
  - Easy to search
  - Focused on the user's task
  - List concrete steps to carry out
  - Not too large

Phases of Heuristic Evaluation

1) Pre-evaluation training
   - Give evaluators needed domain knowledge and information on the scenario

2) Evaluation
   - Individuals evaluate and then aggregate results

3) Severity rating
   - Can do this first individually and then as a group

4) Debriefing
   - Discuss the outcome with design team

How to Perform Evaluation

- At least two passes for each evaluator
  - First to get feel for flow and scope of system
  - Second to focus on specific elements
- If system is walk-up-and-use or evaluators are domain experts, no assistance needed
  - Otherwise might supply evaluators with scenarios
- Each evaluator produces list of problems
  - Explain why with reference to heuristic or other information
  - Be specific and list each problem separately

Examples

- Can't copy info from one window to another
  - Violates "Minimize users' memory load" (H1-3)
  - Fix: allow copying
- Typography uses mix of upper/lower case formats and fonts
  - Violates "Consistency and standards" (H2-4)
  - Slows users down
  - Probably wouldn't be found by user testing
  - Fix: pick a single format for entire interface

Severity Ratings Example

1. [H1-4 Consistency] [Severity 3][Fix 0]

The interface used the string "Save" on the first screen for saving the user's file, but used the string "Write file" on the second screen. Users may be confused by this different terminology for the same function.

Questions?

Summary: HE is a discount usability method

- Based on common usability problems across many designs
- Have evaluators go through the UI twice
- Ask them to see if it complies with heuristics
- Have evaluators independently rate severity

Outline

- User testing process
- Severity and Cost ratings
- Discount usability methods
- Heuristic evaluation
- HE vs. user testing

Outline

- User testing process
- Severity and Cost ratings
- Discount usability methods
- Heuristic evaluation
- HE vs. user testing

2/15/2006

HE vs. User Testing

- HE is much faster
  - 1-2 hours each evaluator vs. days-weeks
- HE doesn't require interpreting user's actions
- User testing is far more accurate (by def.)
  - Takes into account actual users and tasks
  - HE may miss problems & find "false positives"
- Good to **alternate** between HE & user testing
  - Find different problems
  - Don't waste participants

Results of Using HE

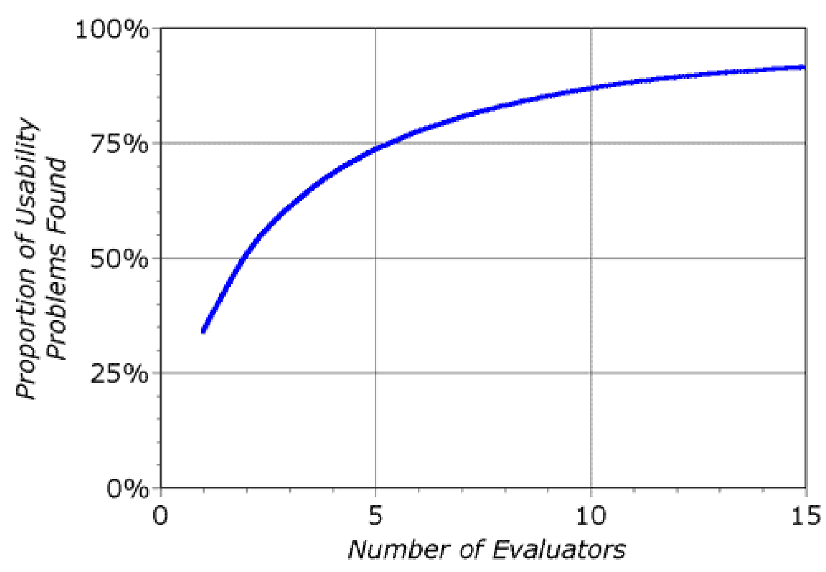
- Discount: benefit-cost ratio of 48 [Nielsen94]
  - Cost was $10,500 for benefit of $500,000
  - Value of each problem ~15K (Nielsen & Landauer)
- Tends to find more of the high-severity problems

Number of Evaluators

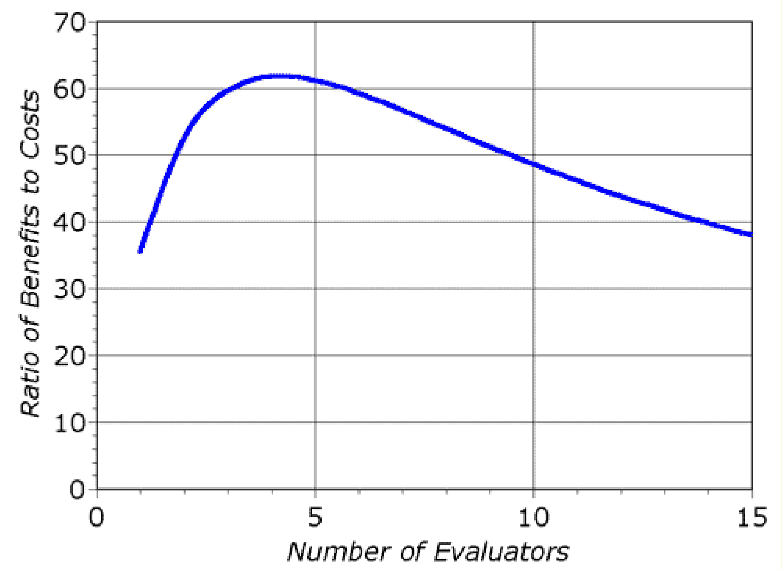
- Single evaluator achieves poor results
  - Only finds 35% of usability problems
  - 5 evaluators find ~ 75% of usability problems
  - Why not more evaluators???? 10? 20?
    - adding evaluators costs more
    - many evaluators won't find many more problems
- But always depends on market for product:
  - popular products -> high support cost for small bugs

Decreasing Returns

problems found



benefits / cost



- Caveat: graphs for a specific example

Questions?

Summary: HE vs. user testing

- User testing is more accurate
- For HE, Combine findings from 3 to 5 evaluators
- Discuss problems with design team
- Alternate with user testing