Evaluation

- Formative vs. summative
- Four paradigms
  - Informal feedback
  - Walkthroughs
  - Field studies
  - Predictive evaluation
- Goals: find problems or new opportunities, check conformity with guidelines, standards, requirements, …

Evaluation planning

- Determine high-level goals
- Explore questions to be answered
- Choose evaluation paradigm and techniques
- Identify practical issues
- Decide how to handle ethical issues
- Evaluate, interpret, and present results

Designing a study

- Reliability: results are repeatable
- Validity: measuring what you want to measure
- Biases: should not be introduced by process
- Scope: breadth of findings' applicability

Interviews

- Structured/scripted vs. unstructured/open-ended
- Avoid long, compound questions
- Avoid unfamiliar terms
- Avoid questions that embody assumptions
- Avoid biases
- Intro, warm-up, main body, cool-off, closing

Questionnaire development

- Paper vs. electronic, closed vs. open-ended
- Checkboxes, rating scales, prose responses
- Design
  - Start off-line even if goal is electronic
  - Questions all positive, all negative, mixed
  - Pilot-test questions for clarity, sufficient space
  - Consider analysis

Increasing questionnaire response

- Expect 20%–40% rate (less online)
- Make purpose clear
- Promise anonymity
- Design well
- Offer short version
- Provide stamped return envelope
- Follow up
- Provide incentive

Expert critiques

- Heuristic evaluation w/ guidelines (Nielsen)
  - Brief 3–5 experts
  - Each works separately 1–2 hours, two passes
  - Debrief experts together
- Cognitive walkthrough
  - Tell expert assumptions, context, task
  - Expert walks through prototype w/ usage scenarios
  - Will user know what to do? Will user see correct action is available? Will user understand response?

Usability walkthroughs

- Make an explicit test scenario (test plan)
- Test the test (pilot study)
- Recruit subjects
- Conduct test
- Debrief subjects

Roles in walkthroughs

- Greeter gets user settled
- Facilitator talks to user during testing
- Computer (a person) manipulates interface elements
- Observer(s) take notes

Observing users

- In the lab
  - Walkthroughs with low-fi prototypes
  - Instrumented sessions with higher-fi systems
- In the field
- Consider, as always, who's involved, their goals, their actions, their feelings, the relevant objects and events

User testing

- A part of usability testing
- Smaller-scale, less formal, more focused than full-blown usability research
- Can be quantitative: time to complete, number of errors, number of help requests, number of users completing task successfully
- Can include keystroke-level monitoring
- How many users?

Le mieux est l’ennemi du bien.
(The best is the enemy of the good.)

—Voltaire

[“Dramatic Art” in *Philosophical Dictionary,* 1764]

Evaluation exercise

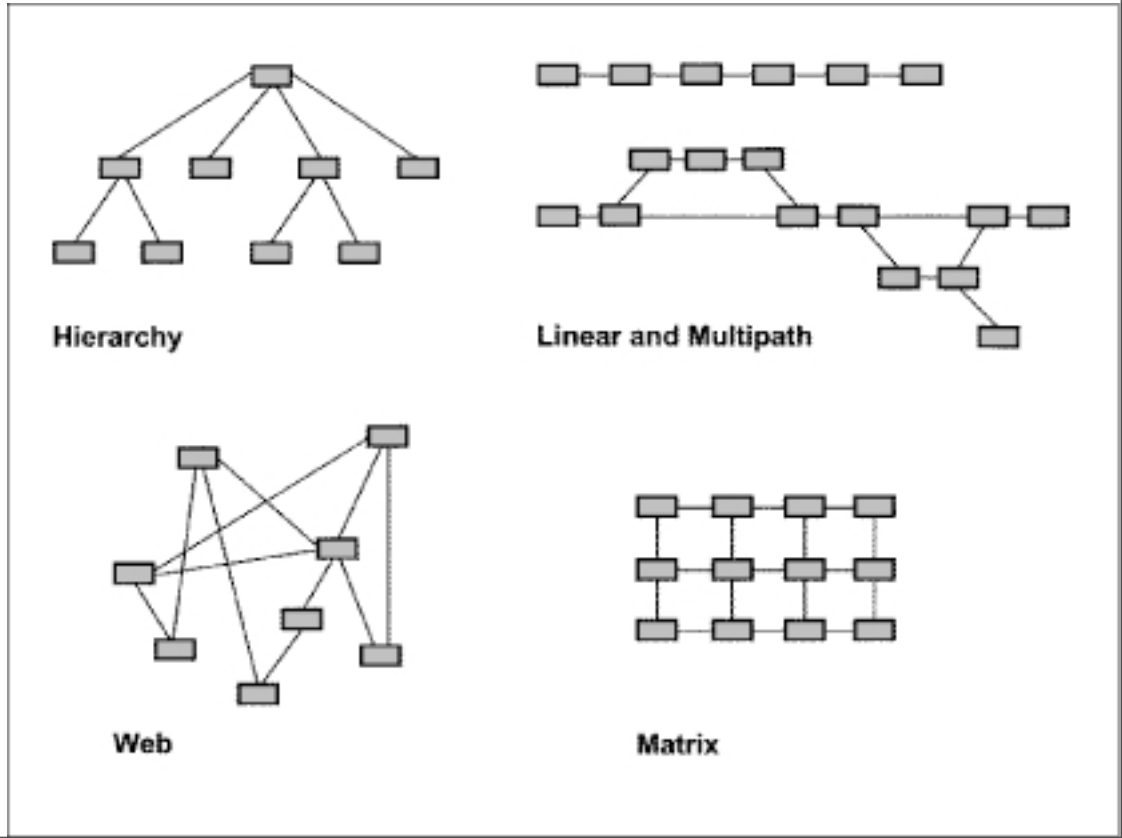
- Start with your ticket reservation system
- Decide on the one aspect that most needs testing/evaluation
- Design an evaluation plan to test that aspect (give specific task, technique, design)
- Describe your plan on one page (informally, possibly with outline or sketches)
- Present to another group for critique (9:20)
- Turn in page with names of group

Hypermedia and the WWW

- Nodes (info in many media)
- Visible links to other nodes
- HCI view: navigation between pages, information presentation, multimedia layout
- More than just HCI design • • • •

Hypermedia organization

- Shape can be hierarchical, linear/multipath, network/web, matrix
- Unrestricted linking makes orientation hard
- Network/web structures hard for beginners



HCI for Web (Farkas & Farkas)

- 1.1: All links indicate they're links •
- 1.2: Help viewers notice links
- 1.3: Links clearly indicate destinations
- 2.1: Effective breadth and depth in hierarchies •
- 2.2: Add secondary/shortcut links where approp.
- 2.3 Allow branches to converge where approp.
- 2.4 Reveal underlying information structure •

HCI for Web (Farkas & Farkas) 2

- 3.1: Clear, conspicuous orientation at top •
- 3.2: Support exploration • • • •
- 4.1: Use site maps for structure and direct access • • •
- 4.2: Provide search facility or index for direct access
- 4.3: Provide links to home page throughout

“Information scent”

- Link should “smell right” to user: confidence before clicking, feel closer afterwards
- Practical measure (Spool 1998):
  - Ask users before clicking what they think they’ll get
  - Ask how confident they are (–2 to +2)
  - Ask users after clicking if they felt closer (–2 to +2)
  - Add the two figures
  - Accumulate those sums as you go from link to link; the result should keep increasing

Advance information

- What's possible now? What will happen next? What can I do now?
- Prevent errors, unexpected results
- Guidelines
  - Give visual indicators, not just text •
  - Distinguish unselectable menu items, objects
  - Change cursor shape •
  - Show submenus on rollover •
  - Show data entry format • •
  - Indicate long operations, ask permission •

Feedback

- User action, system reaction (ideally < 0.1 sec)
- Guidelines
  - Highlight items on rollover
  - Mark selected items •
  - Show path in navigation hierarchy
  - Report errors immediately
  - Use status or progress indicators
  - Use visual, auditory, and tactile modes
  - Make reaction time uniform

Undo

- Encourages users to explore functionality
- Guidelines
  - Special-purpose undo (e.g., backspace) supplements general
  - Try to make everything undoable (external effects clearer to users than internal)
  - Multiple undo (undo/redo or linear sequence)

Error avoidance

- Provide advance information
- Keep dangerous controls away from frequently used ones
- Warn users of irreversible effects; don’t make them the default; request confirmation
- Turn safety options on by default
- Recognize errors and react ASAP

Error messages and actions

- Explain nature of problem, how user can solve it (at least with correct examples)
- Describe in terms of user's task
- Use polite language • • •
- On crash, give opportunity to save
- Support force quit and relaunch

Shneiderman's error message guidelines

- Avoid "fatal," "invalid," "bad"
- Avoid ALL CAPS, cryptic numbers
- Give control over audio feedback
- Give precise messages
- Provide context-sensitive help