Data analysis

• Don’t let data get stale
• Do this iteratively, too
• Decide which tools, how much formalism
  – Scenarios (narrative)
  – Use cases (describe interaction with system, alternative paths)
  – Essential use cases (more abstract: user intention, system responsibility, …)
  – Hierarchical task analysis
Generating alternatives

• No automatic way to come up with ideas
• What kind of interaction (instructing, conversing, manipulating, exploring)?
• Look at similar systems, at very different systems
• Build up your repertoire, your toolbox; expose yourself to a lot of things.
• Techniques: brainstorming, attribute listing and variation, …
Prototyping

- Present ideas for evaluation without getting in too deep (in time, money, commitment)
- Use sketches, storyboards, slide shows, video simulations, physical objects, mock-ups, skeleton software
- Build model of work flow, task design, screen layout, information display, difficult or critical aspects
High-fidelity prototyping

- Same materials as final product
- Realistic-looking results
- Tools include MacroMedia Director, Dreamweaver, VB, …
- Users’ expectations and focus?
Low-fidelity prototyping

- Unlike the final form
- Quick, cheap, easily changeable
- Examples
  - Sketches
  - Index cards
  - Storyboards
  - Sticky notes
- Paper prototyping • • •
Prototyping considerations

• Models necessarily omit detail
• Horizontal vs. vertical approach
• Other tools
  – Denim system (sketches with hyperlinks)
  – Scripting languages (e.g., Tcl/Tk)
User-centered design

- Early focus on users (cognitive, behavioral, attitudinal characteristics) and tasks
- Actual measurement: observe, record, analyze users’ reactions and performance
- Iterative design: find problems, fix them, test again
- Users’ involvement in process
User-centered design

• Affects product acceptance and success
• Makes users active stakeholders
• Manages expectations
• Gets head start on training
• Communicates without sales hype
• Provides vital information about needs, requirements, usability
Time to try it

• Design a web-based system for reserving movie or theater tickets
• Don’t be constrained by existing systems
• Pair up: you will both be users and both be designers
• Determine your context, requirements, tasks
• Design two alternatives with (low-fidelity) prototypes