Prototyping

• What prototyping is
• The benefits of prototyping
• Low-fidelity and high-fidelity prototypes, and the advantages of each
• How to build paper prototypes (storyboards)

Why Prototype?

• Traditional software development: you can’t test until you implement
• Implementation is expensive
• Result: any design errors are built in to the first thing you can test, and it is very expensive to make changes
• Result: design errors, unless they are really bad, are left in the product (as “features”)

Breaking this implementation paradox

• Build a prototype of the basic functionality, especially the interface
• Test the prototype, which will uncover design errors
• Correct the errors
• Repeat until you have a clean design
• Prototyping is
  • a major tool for improving usability
  • Heavily used in industry

The views of the stakeholders

• Software designers may not adequately understand prototyping
• HCI practitioners may not adequately understand implementation
• The two groups need to talk to each other early, and prototyping facilitates communication
Two types of prototypes

- Evolutionary: the prototype eventually becomes the product
- Revolutionary, or throwaway: the prototype is used to get the specs right, then discarded!

Horizontal prototype: broad but only top-level

Vertical prototype: deep, but only some functions

Benefits of prototyping

- Can be used to test every detail of the final product before the product is built (e.g., MoS testing rooms)
- Results in higher user satisfaction
- Users are better at evaluating an existing (vs. described) system
- It brings the users into the process early

Disadvantages

- Users may be unfamiliar with the technique.
- Management may think that the project is nearly finished if the prototype is "too good," or that the prototype can be converted into the final product.
The right way: use low-fidelity prototypes

- Inexpensive
  - In materials cost, people time, and schedule time
- No risk of being mistaken for the final product
- Simple and fast to repeat as lessons are learned
- When interface testing of the prototype is complete, implementation can proceed with confidence

Storyboards: paper-based prototypes

- Navigation
  - visual display of paths
- Interactivity
  - all on paper and words
- Screen design
  - basic layout, basic color
- Rough sketches
  - for key frames, menus, etc.

Storyboarding is about conceptual thinking, not art.

Organization on a single snapshot
- Shows screens, files, concepts, navigation

Screen-level detail makes implementation easy and unambiguous
Thinking out of the box:
You can actually use the prototype!
There is a very high correlation between storyboard quality and final project quality.