User-Centered Website Development: A Human-Computer Interaction Approach
Chapter 7: Prototyping

Copyright © 2004 by Prentice Hall
Credits

- Slides 20-25: DePaul University students:
  - Roy Calayo
  - Marek Dzik
  - Dan Jung
  - Connie Pappas
  - Jessica Watson

- Slides 27-30: City College of New York students:
  - Ashford Charleus
  - Shahana Dewan
  - Khairul Kabir
  - Libai Qu
7. Prototyping

In this chapter you will learn about:

- What prototyping is
- The benefits of prototyping
- Low-fidelity and high-fidelity prototypes, and the advantages of each
- How to build paper prototypes
7.2 Why Prototype?

- Traditional software development: you can’t test until you implement.
- Implementation is expensive, and there is nothing to test until you have made that expenditure of effort and schedule time.
- 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.
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
- 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
The stages in website development

- Design
- One or more prototypes, each followed by testing and redesign
- Implementation
- Site goes live
Two types of prototypes

- Evolutionary: the prototype eventually becomes the product
- Revolutionary, or throwaway: the prototype is used to get the specifications right, then discarded
Horizontal prototype: broad but only top-level
Vertical prototype: deep, but only some functions
Benefits of prototyping

- Improves the chances of creating a usable product, since mistakes and omissions are flushed out early in the development process, not after implementation.
- Higher user satisfaction.
- Users are good at criticizing an existing system; they are much less able to imagine how specifications would translate into a product.
- Prototyping brings the users into the process at a much earlier stage.
Benefits were recognized two decades ago

Implementation is *expensive*

- So expensive—in people time and schedule time—that in practice it is only done once
- If testing is postponed until implementation is complete, there are no resources and no time to do it over when an interface design mistake is uncovered
“The vast majority of usability problems come from a single cause: the development team was lacking a key piece of information. If team members had known it earlier, they would have designed to accommodate it and the usability problem would never have occurred.”

Disadvantages

- Users may be unfamiliar with the technique. But this situation is improving.
- If the prototype is “too good,” management may think that the project is nearly finished, 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
Elements of a paper prototype

- **Menu Bar**
- **Scroll Bar**
- **Secondary Menu**
- **Opening Contents**

**PROLINE GOLF**
Professional Golf Outfitters
500 Golf Road
Arlington Heights, IL

- **Weekly Specials**
  - Golf Tours
  - Golf Woods
  - Metal Driver
  - Golf Shoes
  - Putters

- **Customer Service**
  - Phone Support
  - E-mail
  - Return Policy

- **Information**
  - Golf Course Links
  - Weather Links
  - Vacation Links
  - Golf Instruction
Following are some pages by students

Thanks to Team Cairo, HCI 310, Fall 2001, DePaul University:

- Roy Calayo
- Marek Dzik
- Dan Jung
- Connie Pappas
- Jessica Watson
They started with an 18 x 12 drawing pad
Their home page
User “clicks on” (points to) Clubs button
The Music page

- Music
- Drum-N-Bass
- Hip-Hop
- House
- Techno
- Trance
One Club page

Red NO. 5

- Club Info
- Cover Charge
- Directions
- Dress Code
- Hours

Back
The Club Info page for Red No. 5

Rednove
480 N. Halsted St. map what's nearby
312-733-6699

If Chicagoan John Hughes was still making movies like "Pretty in Pink," he'd be all over Rednove for its dingy, unfinished, '80s rock club ambience. But he's not, and it's not the '80s, and Rednove is not a rock club. It's a dance club, and the result is a strange incongruity between its low-fi upkeep and sensual aspirations. The club consists of a sparse lounge near the entrance and a dance floor/seating area further in. Flanking the geometric, porous walls in the lounge are low couches and ottoman-shaped Seats. Despite the proliferation of colored neon disco lights in Rednove, the dance floor is fairly brightly lit most of the time.

Rednove's upper level, the Fifth Floor, recently underwent a mild facelift and has added a VIP space.

Cost: $20.00
A few pages from another student group

Thanks to Team 2, Computer Science 473, Spring 2003, City College of New York

- Ashford Charleus
- Shahana Dewan
- Khairul Kabir
- Libai Qu
The home page

Pulldown menu
A second-level page
Another second-level page
After prototyping and user testing, this is what their home page looked like
Materials for building a paper prototype

- White, unlined heavy paper or card stock; 11 x 14 or 12 x 18 inches are good sizes
- Regular 8.5 x 11 unlined paper
- 5x8 index cards, for taking notes
- Colored paper if you wish
Materials, continued

- Adhesives: rubber cement, Scotch® tape, glue sticks
- Markers of various colors
- Sticky note pads, in various colors and sizes
- Acetate sheets—a few
- Scissors
Evaluation of your prototype

- Evaluation is the subject of the next chapter
- There you will learn
  - The testing setup
  - Roles of people involved in testing
  - Procedure for testing
  - Interpreting the results
  - Reporting the results
8. Evaluation

In this chapter you will learn about:

- The benefits of testing
- The differences between expert-based and user-based testing
- The proper technique for conducting a user-based test
- Effective means of communicating test results
The benefits of testing

- Usability sells, if the user has a choice—which is almost always
- The designer is a poor choice to test her own site, because she knows too much (Doesn’t have to search for buttons, because he put them there)
- Parallel with software development: no matter how carefully you planned, would you ship a product that had never been tested?
Test early and often

* In traditional software development, users are brought in only at the beta test stage.
* But by then most of the budget has been spent.
* It is very much more expensive to correct an error than if it had been caught early.
* Jared Spool: Bring in two users every week, throughout the development. You uncover lots of errors early. Then do full-scale testing as you near completion.
Formative vs. summative evaluation

- Formative: during development
- Summative: at completion
- “When the cook tastes the soup in the kitchen, that’s formative evaluation; when the guests taste the soup at the dinner table, that’s summative evaluation.”
Test early

- You’re designing and building a house. Compare the cost of moving a bathroom:
  - When you’re looking at the architect’s drawing, before anything has been built
  - After concrete floors have been poured
  - When the walls are plastered and painted, and you’re ready to move in

- Writing your XHTML, Cascading Style Sheets, and JavaScript (or other) isn’t quite like pouring concrete, but it’s close
Expert-based evaluation

- Why bother with user testing? Aren’t there experts who can look at your site and identify problems?
- To an extent, yes. There are experts, and this is done.
- But usually too late. (“We’re going live in two weeks; do you have time to look over our site?”)
- And the expert doesn’t have the characteristics of your users, whom you studied so carefully before starting
Testing with paper prototypes

- As per Chapter 7
- Need a test scenario
Test scenario should state:

- Motives for performing the work
- What the user will be asked to do (actual data rather than generalities)
- The state of the system when a task is initiated
- Readouts of displays and printouts that the test users will see while performing the task
Sample

- Motivation and end results: “Find a woman’s blue V-neck sweater for under $80.”
- State of system: Test user is at the site’s home page. First-time visitor; no data on file. Shopping cart is empty.
- Displays include:
  - Home page
  - Ladies Apparel Department page
  - Sweaters page
  - Search dialog (in case test user decides to search for item rather than clicking on links)
  - List of available sweaters that meet search criteria
Create a prototype, as in Chapter 7
Preliminaries

- Practice with a friend—but don’t include these results as part of the actual testing
- Recruit users
- Preferably not family or friends, because they normally will be trying not to offend you
- If must use family and friend, say something like, “You’ll be doing me a favor by finding mistakes here”
Ideal layout for paper prototype testing
Roles: greeter

- Explains purpose of test
- Makes clear who is present, whether visible to user or not
- Says, “You are not being tested; the product is” (or some equivalent)
- Gets Informed Consent signed
- Offers refreshments
- At end, thanks user, pays ($$, cookies, T shirt)
Informed consent

- See text for one possible form
- Your organization may have a prescribed form
- Main points to include:
  - General purpose
  - Participation is voluntary
  - Results will be confidential
  - There is no benefit to you, other than agreed-upon payment
  - There is no risk to you
  - 18 or over
  - Signature and date
Roles: facilitator

- Only person who speaks to user during test
- Main job is to keep the user talking. This is the “Think Aloud” mode discussed in Chapter 3
- User gets stuck, or stops talking. Don’t give clues, but:
  - What are your options?
  - What are you considering doing?
  - If user asks for help, reflect the question back rather than answering the question
  - At last resort, just say, “That’s fine. Let’s move on.”
Facilitator, continued

- Neutral demeanor at all times
- No signs of impatience: sighing, tapping pencil
- Never criticize, and think twice before praising
- You want to user’s attitude to be, “How can I find that sweater?” not, “What can I do to please the facilitator?”
- Let user struggle until totally stuck
- “Never complain; never explain.” If the interface requires explanation, you have learned that it is deficient.
Roles: “computer”

- In paper prototyping, the person who pulls down the menus, puts a new page in place, and so on
- Goes back to the original (1948) meaning of the word “computer,” which is why ACM stands for Association for Computing Machinery (Machinery? Wouldn’t Association for Computing Machinists make more sense?)
Roles: observer

- Says nothing
- Takes careful notes
- Consider using 3x5 or 5x7 index cards, so they can be sorted in evaluating the test
But . . . you can learn a lot with just one user and yourself

- You can be your own greeter
- Not ideal to combine roles of facilitator and observer, but lots of things aren’t ideal
- Jared Spool says, “Just do it.” Quotes Yogi Berra, “You can learn a lot just by watching.”
Debriefing user: possible approaches

- Open-ended: “What did you like best/least about the site?” What improvements would you suggest?
- Closed-ended: multiple choice, Likert scale, recall of features
- Most of the useful information comes from the notes taken during testing
Using results

- Sort note cards into categories, by type of problem encountered
- Correlate problem areas with prototype, especially site and page navigation
- Look at results in terms of your usability specifications
Refining the design

- If no problems encountered, congratulations!
- Problems may call for redesign
- Easier to convince developers that rework is needed if they watched the test
- Or, if you videotaped, picked out a few sections for a summary of main problems
Writing the report

- Start with an executive summary
- Talk in terms of improvements, not criticism
- Don’t state general design principles; give specifics
- Keep it short
- Prioritize recommendations
- Put testing procedures and raw data in an appendix if at all
Optional: use the NIST Common Industry Format for Usability Test Reports

- NIST = National Institute of Standards and Technology, formerly the National Bureau of Standards
- Usability is so important that vendors and users asked the NIST to devise a common format
- A Word document containing a blank customizable form for a report can be downloaded
- Change the .htm extension to .doc to download the form
Summary

In this chapter you have seen:

- What prototyping is
- The benefits of prototyping: catching design errors *early*, when they are cheap to fix
- The difference between low-fidelity and high-fidelity prototypes, and the advantages of each
- How to build paper prototypes
- Samples of the paper prototypes of two groups of students in HCI courses
Summary

In this chapter you learned:

- The benefits of testing
- The differences between expert-based and user-based testing
- The proper technique for conducting a user-based test
- Effective means of communicating test results
- Usability sells

Usability sells