Rapid Prototyping

Rapid Prototyping in User Interaction Development & Evaluation

Copyright © 2004 H. Rex Hartson and Deborah Hix

Topics

- Relation to usability engineering life cycle and iterative refinement
- Advantages and dangers of prototyping
- Low-fidelity paper prototypes
- Prototype evolution
- What to put in a prototype
- Team exercise on rapid prototyping

Introduction to Rapid Prototyping

- Revisiting the usability engineering life cycle
Rapid Prototyping

- Usability engineering life cycle is evaluation-centered
- Dilemma: Can’t evaluate an interface until it is built, but after building, changes are difficult
- Solution: Rapid prototyping-producing interactive versions of an evolving interaction design

Don’t wait until first release or field test
- Main technique supporting iterative refinement
- Prototype is conversational “prop” to support communication of concepts not easily conveyed verbally (R. Bellamy, Apple Corp.)

Advantages of Rapid Prototyping

- Concrete baseline for communication between users and developers
- Allows users to “take it for a spin”
- Gives project visibility, ‘buy-in’
- Encourages early user participation and involvement
- Allows early observation of user performance
Advantages of Rapid Prototyping

- Example of rapid prototyping advantages (from real world)
  - Database program had novice & advanced users
  - For novice interface spent huge effort and large percentage of system code to build in lots of hand-holding for making queries
  - When released, found that most users moved rapidly from novice to expert, typing in own SQL
  - If could have seen this in advance, could have saved resources and lightened up the application

Advantages of Rapid Prototyping

- Low-fidelity prototyping is obviously not finished, so users have impression it is easy to change
- Allows immediate observation of consequences of design decisions

Advantages of Rapid Prototyping

- Can help sell management an idea for a new product
- Can help effect a paradigm shift from existing system to new system
Dangers of Rapid Prototyping

- Needs cooperation of management, developers, and users
- Management may view as wasteful
- Programmers may lose discipline
- Managers and/or customers and/or marketing may view prototype as final product

Low-Fidelity Prototyping

- Low-fidelity paper prototypes are bona fide techniques on their own
  - Not just a low-tech substitute for computer-based prototype
  - Not just something you do if you don’t have good software tools
  - Major corporations with extensive resources use paper prototype routinely for early interaction development
Low-Fidelity Prototyping

- Paper prototype is essential to support participatory design
- Paper prototypes can evolve very quickly
- Computer-based prototype can distract from usability focus early on
- People do take paper prototypes seriously; they do find many usability problems
- Low-fidelity prototypes find many usability problems, and these are generally the more severe problems

High- & Low-Fidelity Prototyping

- Interaction design has two parts
  - Look and feel: objects
  - Sequencing: behavior, including changes to object behavior

<table>
<thead>
<tr>
<th>Type of Prototype</th>
<th>&quot;Strength&quot;</th>
<th>When in life cycle to apply &quot;strength&quot;</th>
<th>Cost to fix look and feel</th>
<th>Cost to fix sequencing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paper (lo-fi)</td>
<td>Flexibility; easy to change sequencing, overall behavior</td>
<td>Early</td>
<td>Almost none</td>
<td>Low</td>
</tr>
<tr>
<td>Computer (hi-fi)</td>
<td>Fidelity of look and feel</td>
<td>Later</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>

Prototype Evolution Within A Project

- Low-fidelity hand-made paper prototype – evaluate conceptual model, early screen design ideas
- Mid-fidelity prototype – computer printed paper prototype or on-line (VB-like) mockup
- High-fidelity prototype – computer-based with some working functionality (e.g., real database functions)
Prototype Evolution Within A Project

<table>
<thead>
<tr>
<th>Prototype</th>
<th>Evaluation approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenarios &amp; screen designs</td>
<td>Design Walk-Throughs</td>
</tr>
<tr>
<td>Low-fidelity prototype</td>
<td>Usability inspection</td>
</tr>
<tr>
<td>High-fidelity prototype</td>
<td>Lab-based usability testing</td>
</tr>
<tr>
<td>Operational product</td>
<td>Remote usability evaluation</td>
</tr>
</tbody>
</table>

What To Put In A Prototype

- What to put in early prototypes – to evaluate usability of overall interaction design metaphor/paradigm (conceptual design)
  - Low Fidelity
    - Start with representative sample screen or two
    - Mock-up a representative task
    - Follow a representative task thread
  - Learn a great deal from incomplete design, and from a single brand new user

Why Not Go Ahead and Program a Low-Fidelity Prototype?

- You might not have someone on the team who knows VB (or whatever) well enough and who is free to do it
- Paper has much broader visual bandwidth and you often need multiple screens visible at once
- Major reorganization of sequencing faster and easier by shuffling paper on table
Why Not Go Ahead and Program a Low-Fidelity Prototype?

- We often like to write on the pages, screens
- This is a more natural way to edit, modify details during discussion (& always faster)
- Marking paper is far less intrusive to review process
  - Taking 30-60 seconds to make VB change and put back up can break conversation flow, distract focus, and fragment process

Fleshing Out A Prototype

- What to put in later prototypes—to evaluate usability of details (detailed designs)
  - High Fidelity
    - More detailed, refined screens
    - More complete tasks

What To Put In A Prototype

- Invest *just enough* effort in a rapid prototype
  - To achieve formative evaluation goal, but no more
  - Typical transition to real product
- Same process to prototype Web applications as for GUIs
  - Web can make prototype more accessible for different users, conditions
What To Keep From Prototype for Real System

- Keep details (code) of user interaction design objects, look and feel
- Discard any functional or other code holding prototype together
  - E.g., sequencing code; was never meant to be product code

Team Exercise – Rapid Prototyping

Almost all you ever wanted to know about RP you learned in Kindergarten!

- **Goal:**
  - To obtain experience with rapid construction of a low fidelity prototype for early stages of user interaction design

- **Activities:**
  - Draw Web pages or screens to match your scenarios

Team Exercise – Rapid Prototyping

- Make prototypes manually "executable"
- **General – What we are going to do:**
  - Draw interaction objects on paper, cut them out, and tape in aligned positions, relative to other objects, on separate blank plastic transparencies
  - Use "easel to register each sheet of plastics with others sheets"
Team Exercise – Rapid Prototyping

During “execution” most dynamics will be created by adding and removing various registered plastic sheets to/from the easel
- Prototype will be “executed” on the easel, usually taped to tabletop for stability
- You will need to prototype at least the benchmark tasks from your usability specifications, since the prototype will be used in the formative evaluation exercise
- If users can do tasks without any problems, it’s too simple

Team Exercise – Rapid Prototyping

IMPORTANT: Get everyone on your team involved in drawing, cutting, taping, etc. – not just one or two people. You’ll be done much faster if everyone pitches in. However, this is not art class, so do not worry too much about straight lines, exact details, etc.

Team Exercise – Rapid Prototyping

Very important instructions:
- Start with simplest possible background for each Web page in pencil or pen on full size paper, as base for all moving parts.
- Include only parts that never change (e.g., for calendar: monthly “grid”, no month name
- Everything else is drawn in pencil or pen on smaller pieces of paper, cut out, and taped (in proper location) on separate plastic sheet.
- Do not write/draw on plastic, except for transparent objects such as highlights
Team Exercise – Rapid Prototyping

- Don’t draw anything twice; make it modular to reuse
  - The less you put on each layer, the more modular
  - Build up interface in layers
  - We’ll be suspicious of a lot of writing/drawing on one piece of paper
- Whatever changes when user gives input should go on separate paper-on-plastic sheet
- If user will type in values (e.g., item number) use clear sheet on top and marking pen

Team Exercise – Rapid Prototyping

- Make a highlight for major selectable objects
  - Use square or rectangle with “handle”; color with marking pen
  - Make a “This feature not yet implemented” message
- Fasten some objects (e.g., pull-down lists) to top or side of easel with tape “hinges”, so they “flip down” to overlay the screen
- Use any creative techniques to demonstrate motion, dynamics, feedback
  - E.g., scrolling can be done with paper through slits cut in larger paper (all taped to plastic sheet)

Team Exercise – Rapid Prototyping

- Pilot testing: be sure that your prototype will support your benchmark tasks by having one member of your team “run” the prototype while another member plays “user” and tries out the benchmark tasks
- Deliverables:
  - An “executable” version of your prototype, constructed of paper taped in registration to plastic sheets
  - Pilot test complete
- Schedule: Complete by end of class, if possible; else bring to next class