Prototyping

Creation of concrete but partial implementations of a system design to explore usability issues
Goals of Prototyping

*Prototyping enables evaluation, happens throughout*

- Exploring requirements
  - Market analysis, participatory design, envisionment
- Choosing among alternatives
  - Risky or critical features, go/no-go decisions
- Empirical usability testing
  - As early as possible, try out ideas with target users
- Evolutionary development
  - May deliberately choose a malleable software platform, building software in incremental, iterative fashion

*Do scenarios as used in SBD serve as prototypes?*

---

**Boehm’s Spiral Model**
Some Key Tradeoffs

- Quality vs premature commitment
- Special-purpose systems vs scheduling and resource management
- Realism (e.g. timing, content) vs early availability or throw-away efforts
- Constant iteration vs radical change and/or re-factoring of a design
- Dynamic (highly malleable) platforms vs organized, well-structured code base

Prototyping in UE

<table>
<thead>
<tr>
<th>Approach</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storyboard narrative</td>
<td>Sketches or screenshots illustrating key points in a usage narrative</td>
</tr>
<tr>
<td>Paper, cardboard mock-up</td>
<td>Fabricated devices with simulated controls or display elements</td>
</tr>
<tr>
<td>Wizard of Oz</td>
<td>Workstation connected to invisible human assistant who simulates input, output, or processing functionality not yet available</td>
</tr>
<tr>
<td>Video prototype</td>
<td>Video recording of persons enacting one or more envisioned tasks</td>
</tr>
<tr>
<td>Computer animation</td>
<td>Screen transitions that illustrate a series of input and output events</td>
</tr>
<tr>
<td>Scenario machine</td>
<td>Interactive system implementing a specific scenario’s event stream</td>
</tr>
<tr>
<td>Rapid prototype</td>
<td>Interactive system created with special-purpose prototyping tools</td>
</tr>
</tbody>
</table>
“Off-the-Shelf” Prototyping

• Jump-start the design and iteration process
  – Recruit existing tools and devices
  – Integrate into approximation of a “system”
• Example as used in virtual school project
  – Telephone for audio conferencing
  – Netmeeting for video conferencing, chat
  – Web pages for project questions and answers
  – Email for interaction with mentors
• Can be very useful in requirements exploration
  and in activity-oriented feasibility studies

Prototyping Tools

• Presentation tools
  – Paper sketches/printouts
  – PowerPoint
• Scripting languages
  – Tcl/Tk
• Visual languages
  – Visual Basic
  – SILK/Denim

frame $d
button $d.b –text OK
button $d.c –text Cancel
pack $d.b $d.c –side left
Prototyping with Powerpoint

- Create general look-and-feel of interface with essential functionality
- Generate interface widgets using Visual Basic macros
  - Available through toolbar that can be turned on
  - Must set security level to “Low”
  - Actual control functions can only be tested in “slideshow mode”
- Supports creation of an output file for testing

Fidelity in Prototyping

- Fidelity refers to the level of detail
- High fidelity
  - prototypes look like the final product
- Low fidelity
  - artists renditions with many details missing
Why Use Low-fi Prototypes?

- Traditional methods take too long
  - Sketches -> prototype -> evaluate -> iterate
- Can simulate the prototype
  - Sketches -> evaluate -> iterate
  - Sketches act as prototypes
    - Designer “plays computer”
    - Other design team members observe & record
- Kindergarten implementation skills
  - Allows non-programmers to participate

Hi-fi Prototypes Warp

- Perceptions of the tester/reviewer?
  - Formal representation indicates “finished” nature
    - Comments on color, fonts, and alignment
- Time?
  - Encourage precision
    - Specifying details takes more time
- Creativity?
  - Lose track of the big picture
What is SILK?

Sketching Interfaces Like Krazy

Designing Interfaces with SILK

1) Designer sketches ideas rapidly with electronic pad and pen
   - SILK recognizes widgets
   - easy editing with gestures
2) Designer or end-user tests interface
   - widgets behave
   - specify additional behavior visually
3) Automatically transforms to a “finished” UI
Specifying Behaviors

• Sequencing behavior *between* widgets?

  ![Before](before.png) ![After](after.png)

• Storyboards
  – series of rough sketches depicting changes in response to end-user interaction

• Expresses many common behaviors

SILK Storyboards

• Copy sketches to storyboard window
• Draw arrows from objects to screens

• Switch to run mode to test
  – SILK changes screens on mouse clicks
Integrating HCI with Software Construction

- Classic problem in designing from specifications
  - The "specification-design" gap: a written spec is never enough, always ambiguous, always interpreted
  - Who does the interpretation, using what knowledge?
- There are many ways to create tighter linkage
  - OO analysis and design enable simultaneous attention to user task and software design issues
  - Early and continued prototyping is essential
- But, do we want to do this?
  - Only for projects that allow (welcome) requirement shift, that view design as an *inquiry* process
Examples

Tcl/Tk
PowerPoint