MOBILE COMPUTING
CSE 40814/60814
Spring 2017

Specifications
• List of requirements that project/product must meet
• Specifications do NOT state how to build it
• Written document:
  • Do not specify components
  • Written in third person
  • State purpose of project clearly
  • Why are you building it?
  • What will the finished device/app do?
  • Be specific (often tables or drawings)

• Your project proposal is a specifications document!

Specifications
• Examples:
  • Battery life: 6 months continuous use
  • Display:
    • Illumination: visible in strong sunlight
    • Size: min. 4" height and 6" width
    • Resolution: min 800x600 pixels
    • Communication range: 200 feet
    • Weight: max. 15lbs
    • Ruggedness: waterproof to 20 feet; survive 10 feet drop
    • Temperature range: -40 to 120 F
    • Memory capacity: 512MB RAM min.
    • Bootup time: max. 5 seconds
    • Multi-user mode: up to 8 users simultaneously
Specifications

- **Project Reminder:**
  - Project Title
  - Team Members
  - **Problem Statement** (what pain are you addressing?)
  - **Proposed Solution** (what does it do? (high-level))
  - **Technical Details** (what does it do? (detail))

- **Detail:**
  - As much as possible at this point; you will refine/revise this in your project reports and final deliverables

Mobile App Specifications

- **Screens/views**
  - Visual materials, navigation, “look & feel”, portrait/landscape

- **Usability features**
  - Swipe, motion, speech, ...

- **Social media integration**

- **Server integration**

- **Offline work**

- **In-app purchase**

- **Geo-location services**

- **Push notifications**

Sketches vs. Wireframes vs. Prototypes

- **Sketches** are about **exploring** ideas

- **Wireframes & Prototypes** are about **testing** ideas
  - "Low-fidelity representation of design": wireframe
  - "Mid- to high-fidelity representation of final user interface": prototype

- **Process**
  - Sketch: brainstorm design & user experience
  - Wireframe: basic visual guide
  - Prototype: preliminary model (sometimes partially/fully functional)
Sketches vs. Wireframes vs. Prototypes

- **Process**
  - Sketch: explore ideas, brainstorm
  - Wireframe: basic visual guide, structure/flow
  - Prototype: refined structure/flow plus details

- No clear boundaries!!

Sketch vs. Prototype/Wireframe

"Sketching User Experiences", Bill Buxton (...feelings the ends of the continuum should embody...)

<table>
<thead>
<tr>
<th>Sketch</th>
<th>Prototype/Wireframe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invite</td>
<td>Attend</td>
</tr>
<tr>
<td>Suggest</td>
<td>Describe</td>
</tr>
<tr>
<td>Explore</td>
<td>Refine</td>
</tr>
<tr>
<td>Question</td>
<td>Answer</td>
</tr>
<tr>
<td>Propose</td>
<td>Test</td>
</tr>
<tr>
<td>Provoke</td>
<td>Resolve</td>
</tr>
<tr>
<td>Tentative, non committal</td>
<td>Specific Depiction</td>
</tr>
</tbody>
</table>

The primary differences are in the intent

Prototype vs. System Development

- In engineering, prototyping is system development: building the first example of a system by hand
- In user interface design, the effort on the functionality of the system is minimized for the prototype
- Focus on the "visible" parts of the system
- Still a range, in terms of fidelity and level of activity in relation to the final product
What is a prototype?
In designing interactive systems, it can be:
- a series of screen designs (e.g., from Photoshop)
- a storyboard, i.e., a cartoon-like series of scenes
- a PowerPoint slide show or HTML pages
- a video simulating the use of a system
- a lump of wood
- a cardboard mock-up
- a piece of software with limited functionality written in the target language or in another language

Why prototype?
- Evaluation and feedback are central to interaction design
- Users can see, hold, interact with a prototype more easily than a document or a drawing
- You can test out ideas for yourself
- It encourages reflection: important aspect of design
- Prototypes answer questions, and support designers in choosing between alternatives

Low-Fidelity Prototyping (Lo-Fi)
- Wireframing

  • Very far from the final product, e.g., paper, cardboard

  • Examples: sketches of screens, task sequences, etc.
    - Post-it notes
    - Storyboards
    - Scenarios
High-Fidelity Prototyping (Hi-Fi)

- Prototype looks more like the final system than a low-fidelity version
- Common hi-fi prototyping tools:
  - Macromedia Director, Flash, Visual Basic

Lo-Fi vs. Hi-Fi

<table>
<thead>
<tr>
<th></th>
<th>Lo – Fi</th>
<th>Hi – Fi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advantages</td>
<td>Fast</td>
<td>Better sense of finished product</td>
</tr>
<tr>
<td></td>
<td>Cheap</td>
<td>Can judge aesthetic appeal</td>
</tr>
<tr>
<td></td>
<td>Easy – kindergarten skills!</td>
<td>More realistic experience</td>
</tr>
<tr>
<td></td>
<td>Can simulate actual product</td>
<td></td>
</tr>
<tr>
<td>Disadvantages</td>
<td>Slow response time</td>
<td>Users may focus on unnecessary details</td>
</tr>
<tr>
<td></td>
<td>Can’t get feedback about aesthetics</td>
<td>Takes a lot of time to make</td>
</tr>
<tr>
<td></td>
<td>User may question design quality</td>
<td>Users may lose track of big picture</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Great for “big picture”</td>
<td></td>
<td>Great for feel of final product &amp; details</td>
</tr>
</tbody>
</table>

Horizontal vs. Vertical

- How much to represent?
  - “Deep” or “vertical” prototyping
    - provide a lot of detail for only a few functions
  - “Broad” or “horizontal” prototyping
    - provide a wide range of functions, but with little detail
Horizontal vs. Vertical

- Horizontal prototype (e.g., paper prototype)
- Vertical prototype (e.g., Photoshop mock up)

Mobile apps:
- Horizontal: views/screens & flow between them
- Vertical: details of each view

Prototyping Recommendations

- Start early
- Careful with evolutionary prototypes
  - Temptation is too great to stick with bad ideas
- Start with idealistic (rather than realistic) prototypes
- Level of polish should reflect maturity of the prototype

Paper Prototyping

- Easy and fast to do
- Helps you think of specifics
- Usually good as a first round prototype
- Can still do usability testing, even with paper
- Paper Prototyping video:
  - https://www.youtube.com/watch?v=F809lslxjio
Experience Prototyping

- The key is making the interactions and experience as authentic to the real thing as possible
- Typically a hi-fi experience
- Use Wizard-of-Oz (or Oz Paradigm) to save time and avoid complicated/costly implementation

Wizard of Oz

- A method of testing a system that does not exist
  - Simulated listening typewriter (IBM Research 1983)

Dear Henry

What the user sees

http://www.youtube.com/watch?v=NZ6K4EF10pA&feature=related
Important Note

- Up until the point the wizard is discovered, the thoughts, feelings, and actions of Dorothy and the others were all genuine.
- They were genuinely experiencing what it would be like to talk to a powerful and terrible wizard.

Wizard of Oz

- Human ‘wizard’ simulates system response
  - interprets user input according to an algorithm
  - controls computer to simulate appropriate output
  - uses real or mock interface
  - wizard sometimes visible, sometimes hidden
    - “pay no attention to the man behind the curtain!”
- Good for:
  - adding simulated and complex vertical functionality
  - testing futuristic ideas

WoZ Example - Sketch-a-Move

http://www.youtube.com/watch?v=O-Xhewm3Ls8
Prototypes vs. Wireframes

- Prototypes are usually intended to be shown to the end user
- Wireframes are usually more of a design document to go from design to actual system
  - Usually contain annotations specific to the design team and are not intended for end-user consumption
- Wireframes can be used as a lo-fidelity prototype to save time
  - Remove annotations, make it interactive

Example of a Wireframe

Practical Prototyping/Wireframing Tools

- PowerPoint Prototyping
- UX-Specific Tools
  - Axure (website, free for students)
  - Balsamiq (free for 30 days)
  - Mockplus (free; Mockplus Pro $15/month)
  - Moqups (free trial)
  - LucidChart (free trial)
  - Mockingbot (free)
  - Pencil Project (free)
  - Conceptly (free for up to two projects)
  - Fluid (1 project free)
- Photoshop + HTML/Dreamweaver
- Visual Studio
  - OmniGraffle, Gimp
  - Xcode, Android Studio
- Hardware Prototyping (Arduino, Phidgets)
PowerPoint

• Advantages:
  - Almost everyone has it
  - Ubiquitous format
  - Fast and easy to use
  - Can use hyperlinks to simulate interaction

• Disadvantages:
  - Must be used at a computer
    • e.g., difficult to do mobile-based interactions
  - Somewhat limited functionality
  - Cannot be reused for final implementation

Blackboard Wireframing Example

• Fancy Weather App

Back to Project

• Wireframe/Prototype due 2/3 11.59pm
  - Summarize project idea
  - Describe specifications
  - Design wireframe/prototype of your solution
    - App:
      • Show all views/screens with proposed choices and layouts of UI components
      • Show navigation between views/screens
    - Hardware (e.g., no UI):
      • Show architecture & components of hardware
      • Show functionality and/or interactions graphically (e.g., flow graphs)