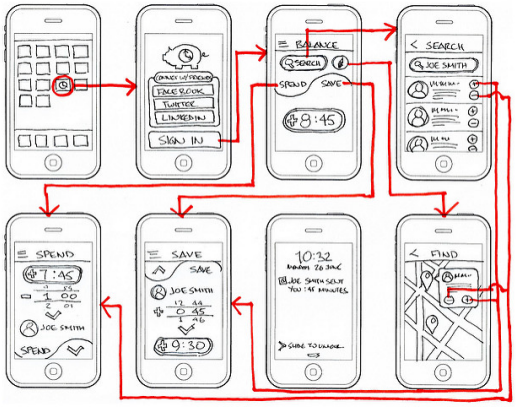
MOBILE COMPUTING

CSE 40814/60814
Spring 2021

Today: Mobile App Development

- Sketches, wireframes, prototypes



Specifications

- **List of requirements that project/product must meet**
- Specifications do NOT state how to build it
- Written document:
  - "formal document used to describe a product's intended capabilities, appearance, and interactions with users in detail for software developers"
  - Do not specify components; focus on WHAT, not the HOW!
  - 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)

Specifications

- Examples (Hardware):
  - Battery life: 6 months continuous use
  - Solar powered; charge time < 6 hours for full charge under cloudy conditions
  - 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

Specifications

- Examples (Software):
  - User registration, login, password recovery
  - Display items by price, reviews, popularity
  - Display users on map
  - Make purchases using Visa, MC, Paypal, Square, …
  - Peak performance: serve up to 10,000 requests per second
  - Storage for 10 million users and 1GB per user
  - Availability of 99.9%
  - Latency < 500ms
  - Notify user of price changes via text or email

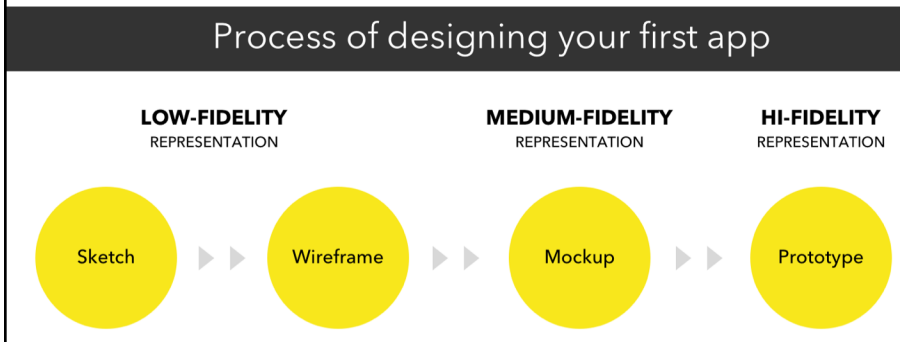
Specifications

- Examples (Mobile App):
  - 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

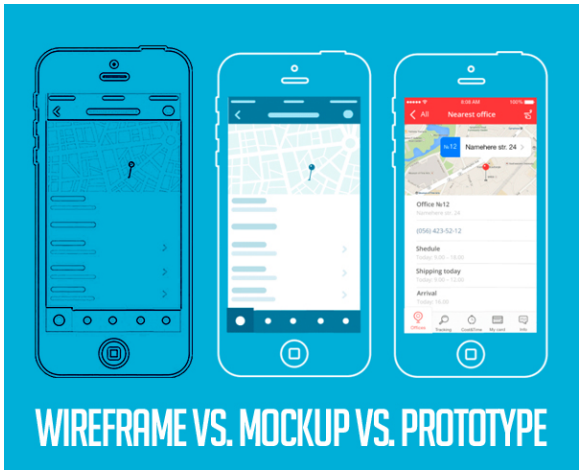
Sketches vs. Wireframes vs. Prototypes

- Sketches are about **exploring** ideas
- Wireframes are about **refining** ideas
  - Low-fidelity representation of design
  - "Mid- to high-fidelity representation of final user interface": prototype
- Mock-ups & Prototypes are about **testing** ideas
  - Mid- to high-fidelity representation of final user interface
- Process
  - Sketch: brainstorm design & user experience
  - Wireframe: basic visual guide
  - Mock-up & Prototype: preliminary model (sometimes partially/fully functional)

Sketches vs. Wireframes vs. Prototypes

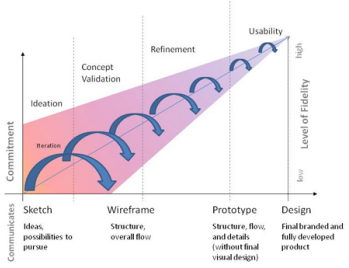


Sketches vs. Wireframes vs. Prototypes



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" by Bill Buxton

Sketch	Prototype/Wireframe
Invite	Attend
Suggest	Describe
Explore	Refine
Question	Answer
Propose	Test
Provoke	Resolve
Tentative, non committal	Specific Depiction

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?

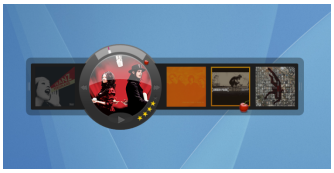
- **Evaluation and feedback** are central to interaction design
- Users can **see, hold, interact with a prototype** more easily than a document or a drawing
- You can **test out ideas for yourself**
- It **encourages reflection**: important aspect of design
- Prototypes **answer questions**, and support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

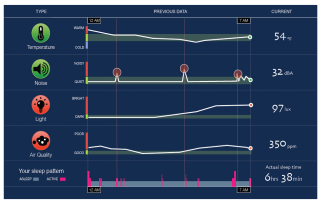
Low-Fidelity Prototyping (Lo-Fi)

- 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

	Lo – Fi	Hi – Fi
Advantages	• Fast • Cheap • Easy – kindergarten skills! • Can simulate actual product	• Better sense of finished product • Can judge aesthetic appeal • More realistic experience • Can evaluate experience
Disadvantages	• Slow response time • Can't get feedback about aesthetics • User may question design quality	• Users may focus on unnecessary details • Takes a lot of time to make • Users may lose track of big picture

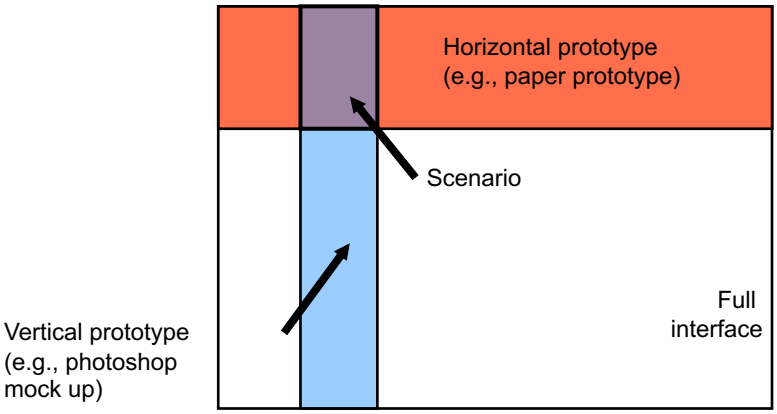
Great for "big picture"

Great for feel of final product & details

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



Mobile apps:

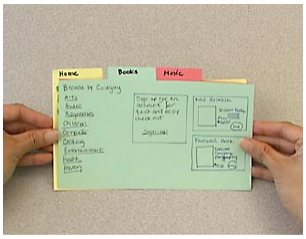
- Horizontal: views/screens & flow between them
- Vertical: details of each view

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=FS00Ulo12Xk

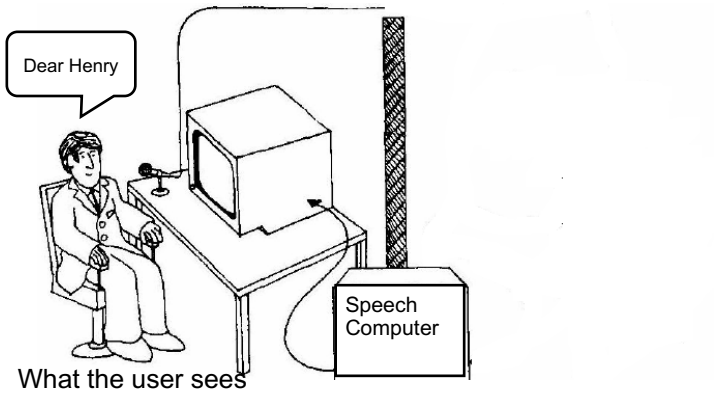


Experience Prototyping

- 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

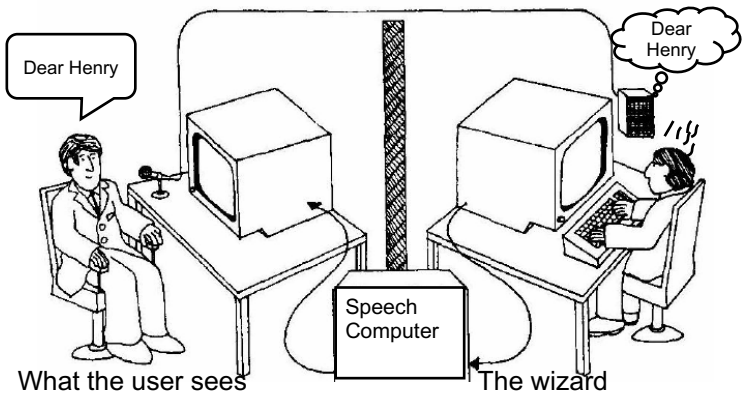
Wizard of Oz

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



Wizard of Oz

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



http://www.youtube.com/watch?v=NZR64EF3OpA&feature=related

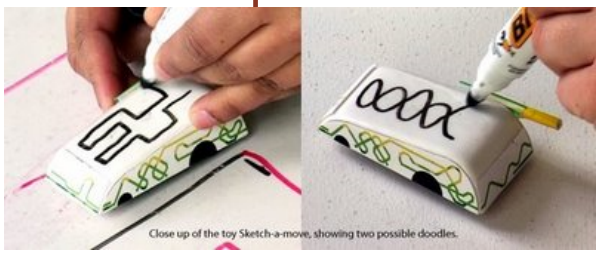
Important Note

- 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



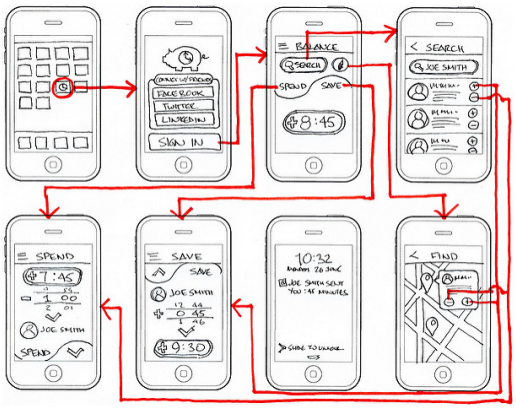
http://www.youtube.com/watch?v=O-XNwam3LOs

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 low-fidelity prototype to save time
  - Remove annotations, make it interactive

Example of a Wireframe

Example of a Wireframe



Practical Prototyping/Wireframing Tools

- PowerPoint Prototyping
  - https://speckyboy.com/free-wireframe-templates-mobile-app-web-ux-design/
- UX-Specific Tools
  - Axure (websites, 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)
  - Concept.ly (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

That's it for today…

- Next up: Mobile app development basics & concepts