

# Prototyping & Executing a Capstone

*How prototyping can help in your capstone,  
especially when done with User-Centered Design...*

## Designing Your Capstone

**Decide which users/tasks/experiences you will support.**

- It might not be practical to design a project to support each and every task/user/experience that you envision.
- To start off you need to determine what is ***needed*** - talk to people, observe them, identify things that might be extraneous or optional, do a “literature review” to see what’s missing in the application ecosystem, etc.

**You then iterate through the following three phases:**

- User/task/experience generation and analysis.
- Forming your ideas into designs.
- Creating prototypes to have users try out (typically on the tasks you’ve developed).

## Designing Your Capstone

**You need to determine how will things appear to the users or audience!**

- This is what they first see – it needs to invite use.

**You'll want to think about what each step through a given task or experience will look like...**

- There should be a natural progression as the user accomplishes their task or audience member has their experience.

**At some point you have a “final spec” for the project (or for a part of the project) that you want to implement for the capstone.**

## Walk through your project concept

**Before you implement something, try to evaluate via a low-fidelity prototype of it.**

- Use the task/experience examples to walk through your design to evaluate whether it will be usable or convey the ideas that you planned.

**For each scenario you have, go through the described task or experience step by step.**

- Is the motivation clear at each step?
- Can you expect the user to know what to do at each step with the anticipated level of prior knowledge?

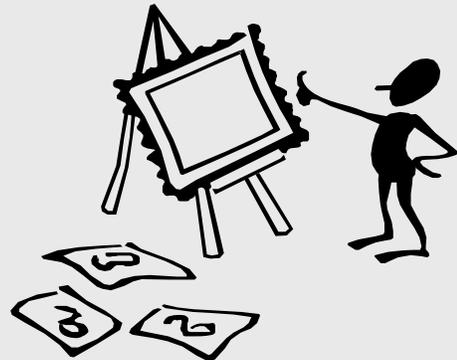
## Low fidelity prototypes

Paper-based prototypes are most common, a paper mock-up of the interface look, feel, functionality. For audio and video, this takes a different form, but can still be paper-based...

Key point: “quick and cheap” to prepare and modify

### Purpose

- brainstorm competing representations
- elicit user reactions
- elicit user modifications / suggestions



Evan Golub / Ben Bederson / Saul Greenberg

## Low fidelity prototypes: Sketches

Drawing of the outward appearance of the intended system.

- crudity means people concentrate on high level concepts
- but hard to envision a the full progression

Generally not good to have too much typed! Should really be hand-drawn on paper.

Evan Golub / Ben Bederson / Saul Greenberg

## Low fidelity prototypes: Iterate

To get a good idea, start by getting lots of ideas...

**The “speed” of lo-fi prototypes makes it fundamentally easier to go through several iterations – each with feedback from users.**

Evan Golub / Ben Bederson / Saul Greenberg

## Low fidelity prototypes: Storyboarding

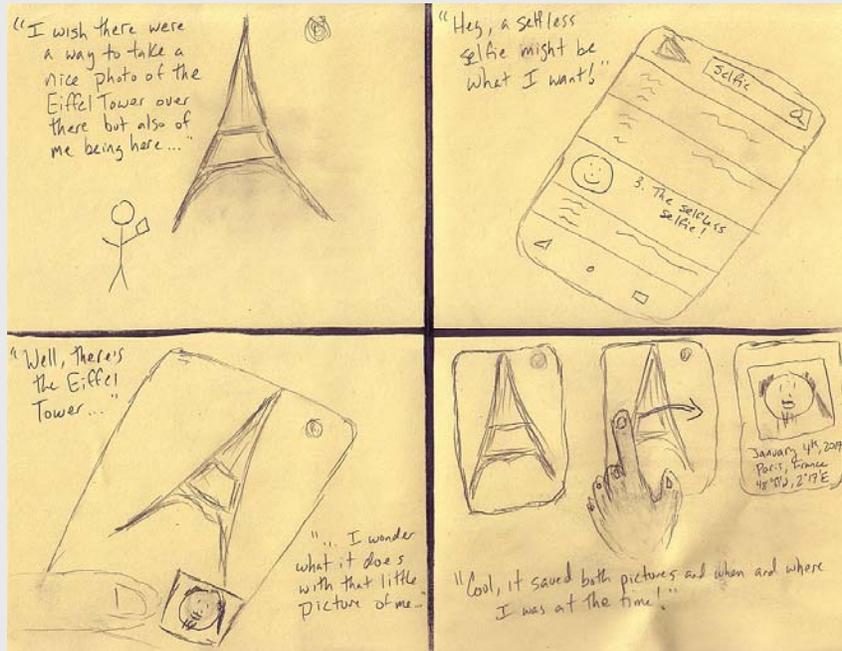
This can be done / thought of as a series of key frames.

- originally from film; used to get the idea of a scene
- can also be snapshots of the interface at particular points in the interaction

The users can evaluate quickly the direction the interface is heading before you write the first line of code!

Evan Golub / Ben Bederson / Saul Greenberg

## Storyboard of using a new type of selfie app



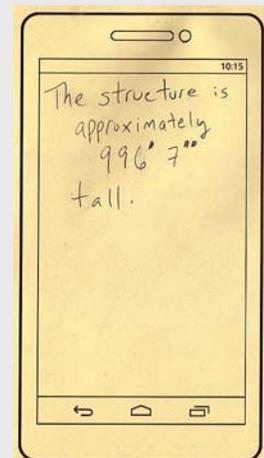
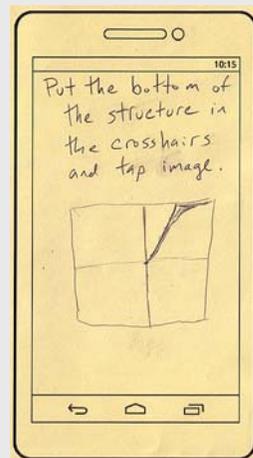
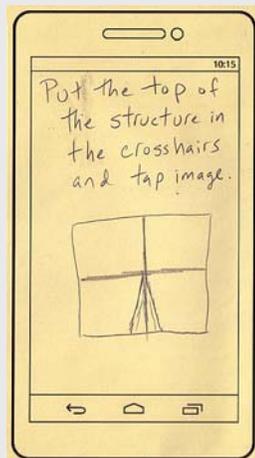
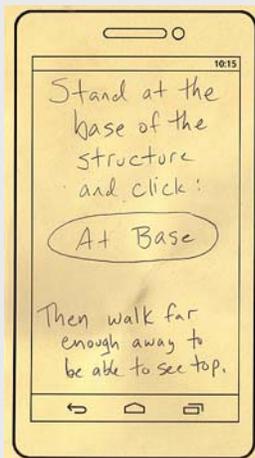
## Storyboard of an app to measure height of a structure

The user taps the button when they are at the base of the structure which records the GPS location and then they step back around a hundred feet.

The user gets the top of the structure dead center and tap the image to record the GPS location and the device's tilt.

The user gets the bottom of the structure dead center and tap the image to record the GPS location and the device's tilt.

The app uses math and science to calculate the approximate height of the structure! We can use trigonometry on the angles and GPS-based distance...



## Low fidelity prototypes: Music

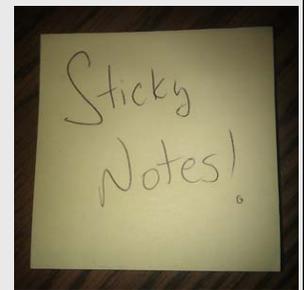
Could lay out the sheet music concept and lyrics and “hum it out” to get a more visceral feel before going in to do more detailed composition and recording...

Evan Golub / Ben Bederson / Saul Greenberg

## PICTIVE prototypes for software/websites

“Plastic Interface for Collaborative Technology Initiatives through Video Exploration” - Muller, CHI 1991

- Design is multiple layers of sticky notes and plastic overlays
  - different sized stickies represent icons, menus, windows etc.
- Interaction demonstrated by manipulating notes
  - contents changed quickly by user/designer with pen and note repositioning
- Session can even be recorded for later analysis
  - usually end up with mess of paper and plastic!



## Playtending

One form of the “original” Palm Pilot as “used” by Jeff Hawkins, carrying it around for months as if it was real to see how it needed to be designed.



See also, the Wii U Gamepad

<http://www.ign.com/articles/2012/11/08/check-out-this-early-wii-u-gamepad-prototype>

## Fail Fast

**Low fidelity tools, such as**

- arts and crafts supplies
- hand-drawn mock-ups
- storyboards
- “screenshots” of widgets
- transparencies
- sticky notes

**allow for rapid iteration with little time or cost (or emotional attachment) and give the users the most freedom to suggest changes.**

**This is sometimes thought of as the “fail fast” stage.**

## Low/Medium Hybrids

### **Photo-based sketches**

Start with a photograph of a real space and sketch in the “new” thing you are working on.

### **More playtending...**

Video “mock-ups in action” to analyze flow...

[https://www.youtube.com/watch?v=x48qOA2Z\\_xQ](https://www.youtube.com/watch?v=x48qOA2Z_xQ)

<https://www.youtube.com/watch?v=-SOeMA3DUEs>

## Medium Fidelity

**After a few rounds of low fidelity brainstorming and feedback, you move on to some form of medium fidelity prototype which is interactive and less rough.**

- Wireframes/flowcharts for more formal planning
- Interactive mock-ups based on flowcharts
- Toolkits for realistic mock-ups
- Specs to get the size of things realistic
- Domain-specific tools
- More coding-centric tools
- Wizard of Oz
- Physical objects

**These are not mutually exclusive things...**

Evan Golub / Ben Bederson / Saul Greenberg

## Medium fidelity prototypes

### **Wireframes/Flowcharts**

- for more formal planning.
- can build interactive mock-ups based on flowcharts

### **Prototyping with a computer**

- simulate or animate some but not all features of the intended system
  - engaging for end users

### **Purposes**

- provide a sophisticated (limited) scenario for the user to try
- provide a development path towards functional system
- can test more subtle design issues

Evan Golub / Ben Bederson / Saul Greenberg

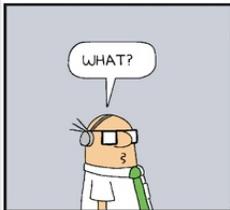
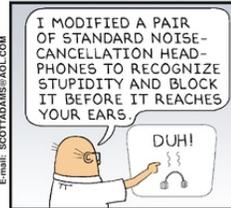
## Medium fidelity prototypes

### **Wizard of Oz - A method of testing a system, or a part of a system, that does not yet exist.**

- human simulates the system's intelligence and interacts with user
- uses real or mock interface
  - “Pay no attention to the man behind the curtain!”
- user uses computer as expected
- “wizard” (preferably hidden):
  - interprets subjects input according to an algorithm
  - has computer/screen behave in appropriate manner
  - might have errors artificially introduced
- good for:
  - adding simulated and complex vertical functionality
  - testing futuristic ideas
- ongoing research into WoO tools (SketchWizard, UISKEI, i2ME)

Evan Golub / Ben Bederson / Saul Greenberg

 **DILBERT**<sup>®</sup> BY **SCOTT ADAMS** 



<http://dilbert.com/strip/2005-04-03>