

## Prototyping & Building a System

*How Prototyping helps (especially when done with User-Centered Design)*

### Prototyping



<http://dilbert.com/strip/2009-07-22>

## Designing Your System

### **Decide which users and tasks you will support.**

- It might not be practical to design a system to support each and every task and/or user that you discovered in the previous stage.
- 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 and Task generation and analysis.
- Forming your ideas into designs.
- Creating prototypes to have users try out (typically on the tasks you’ve developed).

## Designing Your System

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

- This is what the user first sees – it needs to invite use.

### **You’ll want to think about what each step through a given task will look like...**

- There should be a natural work flow as the user accomplishes their task.

**At some point you have a “final spec” that you want to implement as a final product.**

### Quick Activity

**On your sticky note, please draw  
a tree with a house next to it and a cat in it.**

Evan Golub / Ben Bederson / Saul Greenberg

### Prototyping at different project stages...

#### Early design

Brainstorm different representations  
Choose a representation  
Rough out interface style  
Task centered walkthrough and redesign

Fine tune interface, screen design  
Heuristic evaluation and redesign

Usability testing and redesign

Limited field testing

Alpha/Beta tests

Low fidelity paper prototypes

Medium fidelity prototypes

High fidelity prototypes / restricted systems

Working systems

#### Late design

Evan Golub / Ben Bederson / Saul Greenberg

### Low Fidelity Prototyping

**One way to support the involvement of different types of users is to use non-electronic/software solutions.**

- This allows for faster and cheaper iterations of ideas and removes issues of technical failures and debugging.
- It can lower the technical expectations/needs of the team's users, allowing a wider range of participants.
- Without needing to build things, in theory you aren't limited to existing technologies and core ideas can more easily surface.
- It can support storytelling, which helps provide context along with specific ideas.

### Walk through your design

**Before you implement anything, evaluate via a low-fidelity prototype.**

- Use the tasks examples to walk through your design to evaluate whether it will be usable.

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

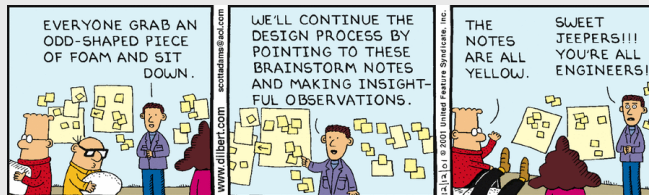
- Is the motivation clear at each step?
- Can you expect the user to know what to do at each step with the anticipated level of training?

### Evaluation Technique: Sticky Note Brainstorming

A question is presented and the team (usually each member but sometimes in pairs) writes one word or phrase on each sticky note.

Session leaders collect the individual notes and organize them on a wall or board, looking to cluster similar ideas together to discover common themes and give a brief name to each.

**This approach works well across many different contexts, but needs to be understood.**



### Technique: Likes, Dislikes, Design Ideas

**This approach also uses sticky notes, but in a guided form and experiential context.**

- A design target is selected and the team spends some time trying out this software or hardware (or even paper) prototype.
- As they use it, every time they see something they like or dislike or they have an idea for a new feature, they write it down (very briefly) on a sticky note.
- Similar to sticky note brainstorming, the notes are collected, but this time they are organized not only by themes, but in columns for Likes, Design Ideas, and Dislikes.

## Low fidelity prototypes

### **Paper-based prototypes**

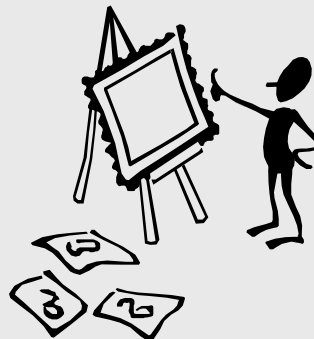
- a paper mock-up of the interface look, feel, functionality
- “quick and cheap” to prepare and modify

### **Purpose**

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

### **Research**

- researchers work on new ways to make and use these prototypes



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 dialog’s progression

~~**Computer Telephone**~~

~~Last Name:~~

~~First Name:~~

~~Phone:~~

~~Place Call~~

~~Help~~

**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

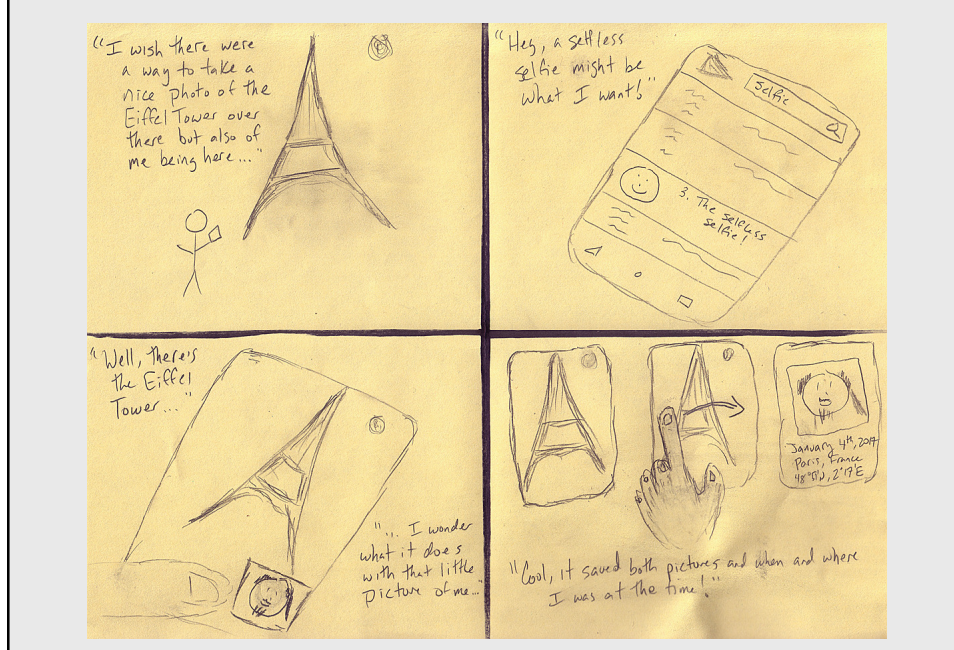
This can be done / thought of as a series of key frames.

- 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



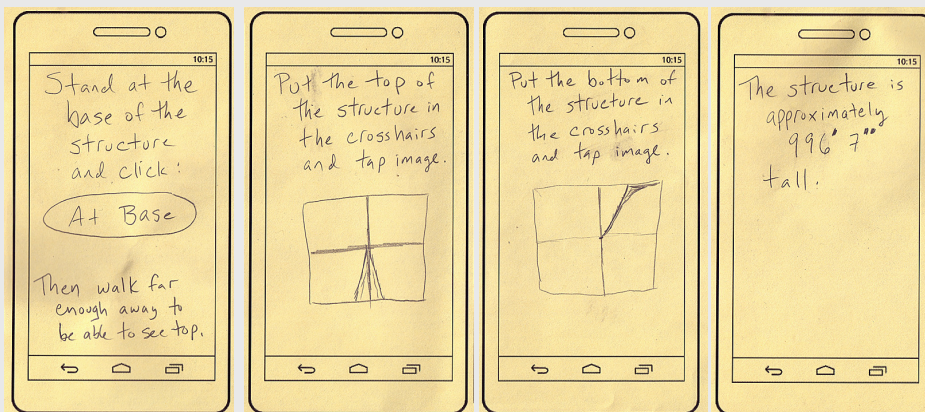
## Storyboard of an app to measure height of a structure

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...

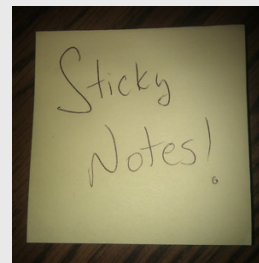




## PICTIVE prototypes

“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!

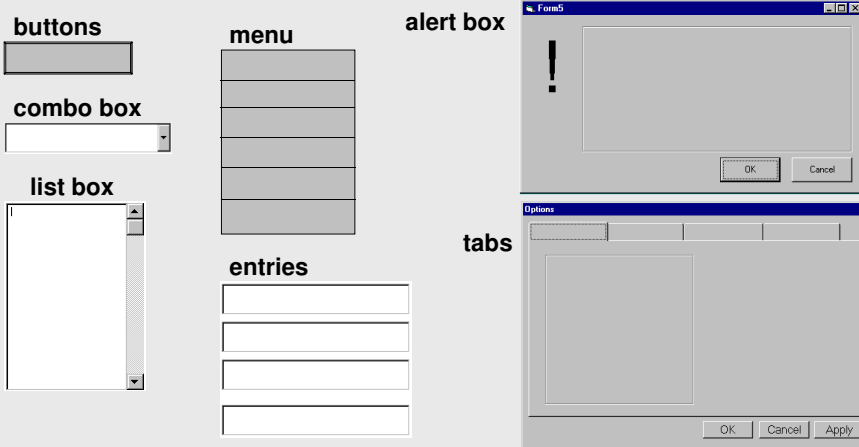


Evan Golub / Ben Bederson / Saul Greenberg

## PICTIVE prototypes

Can create pre-made interface components on paper  
(though this can lock users into a certain mindset).

e.g., these empty widgets were created in Visual Basic and could be printed out:



*I would argue it is still better to hand-draw them...*

Evan Golub / Ben Bederson / Saul Greenberg

### Activity

**Using arts and crafts type supplies, teams will design an HVAC control system for one of several contexts...**

Evan Golub / Ben Bederson / Saul Greenberg

### Fail Fast

**We've talked about low fidelity tools**

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

**These 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.**

## 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

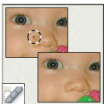


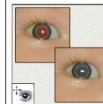

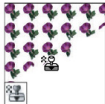
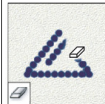








<http://www.ign.com/articles/2012/11/08/check-out-this-early-wii-u-gamepad-prototype>

## Other uses of low fidelity prototypes

### **Tutorials and manuals**

- write them in advance of the system
- what are they?
  - tutorial for step by step description of an interaction
    - an interface “walk-through” with directions
  - manual for reference of key concepts
    - in-depth technical description
- if highly visual, then a storyboard can be set within textual explanations of what the user should be doing

I’m told that people will even sometimes read through the manuals of competing products to check up on their interface, the functionality of the system, and how well these match up with tasks...

 <p><b>The Spot Healing Brush tool<sup>®</sup></b> removes blemishes and objects.</p>	 <p><b>The Healing Brush tool<sup>®</sup></b> paints with a sample or pattern to repair imperfections in a image.</p>	 <p><b>The Patch tool<sup>®</sup></b> repairs imperfections in a selected area of an image using a sample or pattern.</p>	 <p><b>The Red Eye tool<sup>®</sup></b> removes the red reflection caused by a flash.</p>	<p><b>Photoshop CS2 Retouching tools gallery from help</b></p>
 <p><b>The Clone Stamp tool</b> paints with a sample of an image.</p>	 <p><b>The Pattern Stamp tool<sup>®</sup></b> paints with part of an image as a pattern.</p>	 <p><b>The Eraser tool</b> erases pixels and restores parts of an image to a previously</p>	 <p><b>The Background Eraser tool<sup>®</sup></b> erases areas to transparency by dragging.</p>	
 <p><b>The Magic Eraser tool</b> erases solid-colored areas to transparency with a single click.</p>	 <p><b>The Blur tool<sup>®</sup></b> blurs hard edges in an image.</p>	 <p><b>The Sharpen tool<sup>®</sup></b> sharpens soft edges in an image.</p>	 <p><b>The Smudge tool<sup>®</sup></b> smudges data in an image.</p>	
 <p><b>The Dodge tool<sup>®</sup></b> lightens areas in an image.</p>	 <p><b>The Burn tool<sup>®</sup></b> darkens areas in an image.</p>	 <p><b>The Sponge tool<sup>®</sup></b> changes the color saturation of an area.</p>		

■ To open the Memos Folder, click the Open button.

Directory title

Items you can open

Open button

Memos Folder

First Memo

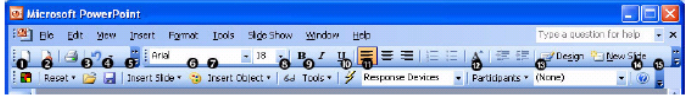
Second Memo

System Tools

As you open the Memos Folder, you move down through the hierarchy. The directory title changes to remind you where you are in the hierarchy, and the box shows you what's on the new level you just moved to—in this case, the two documents in the Memos Folder. The selected document is the one that will open when you click the Open button. If you want to open the other document, click anywhere on the other document's name to highlight it, and then click the Open button.

**1980s - From Apple's Tutorial Guide to the Macintosh Finder**

**TurningPointToolbar**



The following descriptions define the buttons that make up the tools in the TurningPoint toolbar. The numbered labels in the previous image show you the location of each tool.

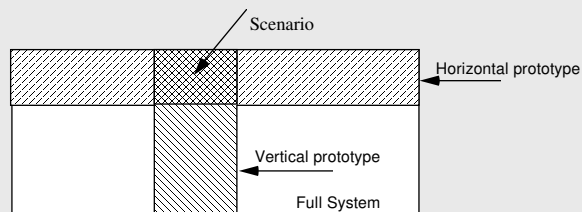
**1** Activate/Update      Allows you to submit the license information for TurningPoint software and upgrades. Find information about installation and customer support in *Chapter 1: About TurningPoint*.

**From the TurningPoint User Guide**

## Incomplete Prototypes

### Approaches to limiting prototype functionality

- vertical prototypes
  - includes in-depth functionality for only a few selected features
  - common design ideas can be tested in depth
  
- horizontal prototypes
  - surface layers includes the entire user interface with no underlying functionality
  - a simulation; no real work can be performed
  
- scenario
  - scripts of particular fixed uses of the system; no deviation allowed



Evan Golub / Ben Bederson / Saul Greenberg

## 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...**

## 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

## Some tools...

**Software that allows you to prototype other software includes PowerPoint, InVisionApp, MarvelApp, Moqups, Balsamiq, Proto.io, Framer.com, Javascript, Flash, Silverlight, HTML5, etc.**

**Physical realism is sometimes needed so you might want to get certain hardware specifications to have the actual size of things be accurate (resources such as <http://screensiz.es/phone> exist).**

**Some domain-specific tools exist, such as “Prototyping on Paper” for iOS (by Woomoo) and physical objects can be useful... cardboard, clay, vinyl, 3D designed/printed, etc.**

### “Dangers” of Medium Fidelity prototypes

**Medium fidelity prototypes might take too long to build and might be hard to change.**

- Reduces number of iterations

**User’s reactions usually get “in the small” at this level.**

- blinds people to major representational flaws

**Developers might be more likely to resist changes.**

- “but it is already working...”

**A single bug can halt testing!**

**Management may think its real!!!**

Evan Golub / Ben Bederson / Saul Greenberg

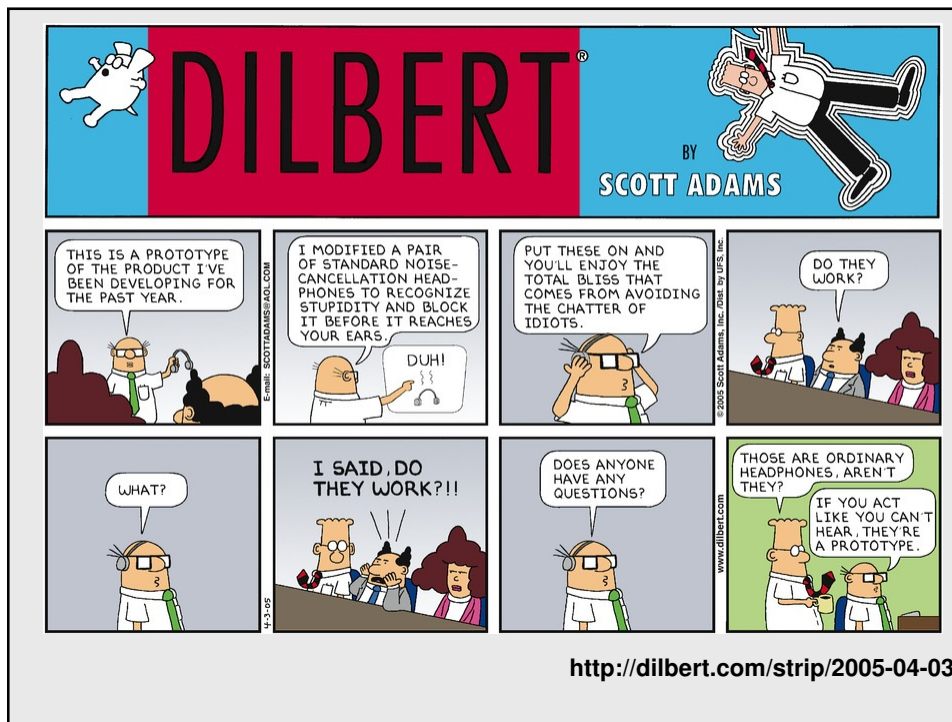
### Medium fidelity prototypes

**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





## Wizard of Oz Examples (I)

### **IBM: an imperfect listening typewriter using continuous speech recognition**

A secretary was trained to:

- understand key words as "commands"
- to type responses on screen as the system would
- manipulating graphic images through gesture and speech

### **Intelligent Agents / Programming by demonstration**

- person trained to mimic "learning agent"
  - user provides examples of task they are trying to do
  - computer learns from them
- shows how people specify their tasks

**In both cases, system very hard to implement, even harder to change!**

## Wizard of Oz Examples (II)

**Imagine scenarios where you aren't sure whether the investment is worth the 'payout' or you want to develop the technology while exploring interface ideas.**

- You want to build a map system that shows where the user is in real-time. Rather than needing to install tracking systems before being able to do the UI testing, you could have a wizard watching the users and updating their location manually on the system.
- You want to have location-aware directional cues such as blinking lights or arrows or sound effects turn on and off as appropriate to guide a user to a destination. Again, you could have a wizard instruct the system to turn things on and off without having the proximity sensors installed or heuristics to determine the user's directional orientation.

Evan Golub / Ben Bederson / Saul Greenberg

## What you now know about...

### **Prototyping**

- allows users to react to the design and suggest changes
- low-fidelity prototypes best for brainstorming and choosing representations
- medium-fidelity prototypes best for fine-tuning the design

### **Prototyping methods**

- vertical, horizontal and scenario prototyping
- storyboarding
- Pictive
- scripted simulations
- Wizard of Oz

Evan Golub / Ben Bederson / Saul Greenberg

## Readings

Required: On ELMS: Rettig, M. (1994) *Prototyping for tiny fingers*. Communications of the ACM, 37(4), ACM Press.

Optional: On ELMS: Jim Rudd, Ken Stern, and Scott Isensee. (1996) *Low vs. high-fidelity prototyping debate*. Interactions 3(1).