Prototyping & Building a System

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

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…

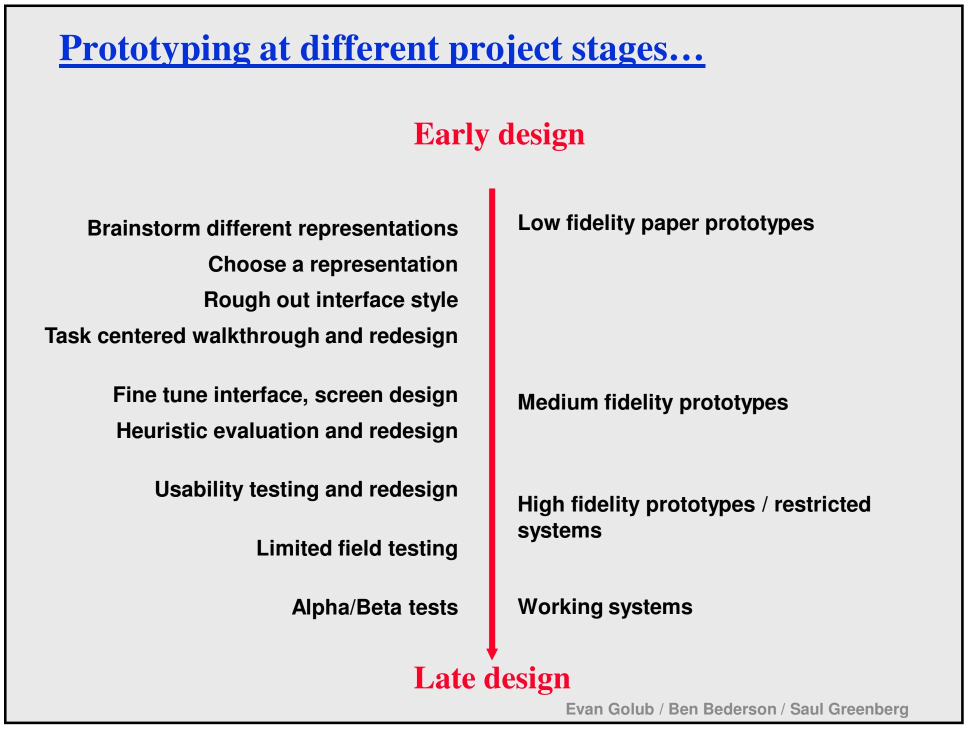
- There should be a natural work flow as the user accomplishes their task.

At some point you have a "final spec" that you want to implement as a final product.

Prototyping

THE PROTOTYPE IS DONE. COME TAKE A LOOK AT THE USER INTERFACE.
www.dilbert.com scottadams@aol.com
IT WORKS GREAT, BUT MAKE SURE THIS THING IS TOTALLY IDIOT-PROOF.
7-22-09 ©2009 Scott Adams, Inc./Dist. by UFS, Inc.
AGAIN?

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



Prototyping at different project stages…

Early design

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

Low fidelity paper prototypes

Fine tune interface, screen design
Heuristic evaluation and redesign

Medium fidelity prototypes

Usability testing and redesign

High fidelity prototypes / restricted systems

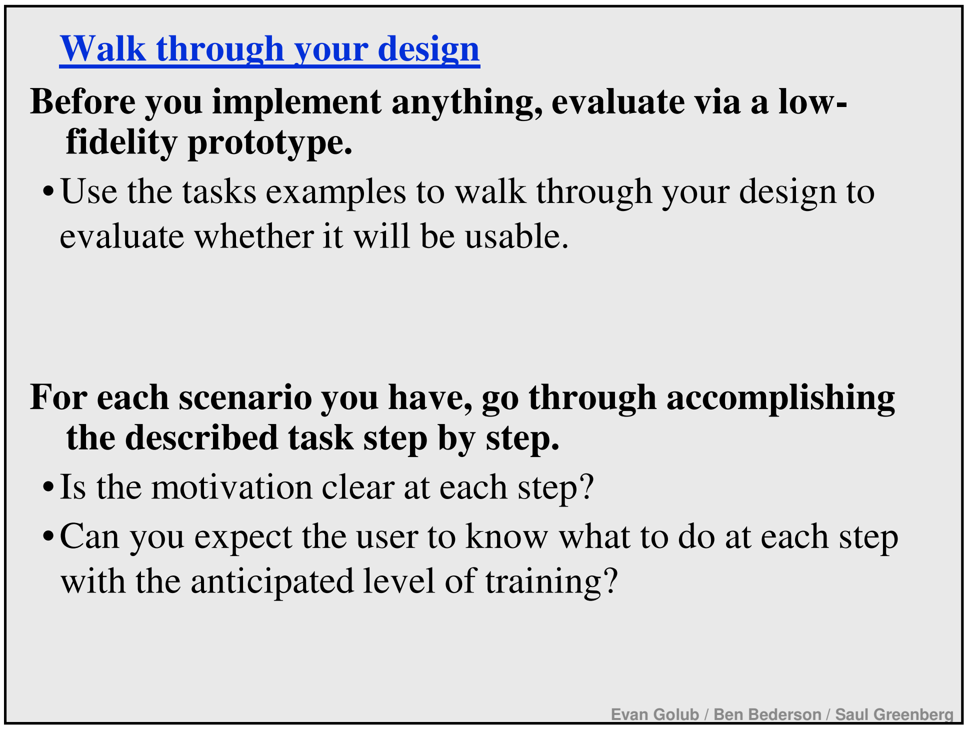
Limited field testing

Alpha/Beta tests

Working systems

Late design

Evan Golub / Ben Bederson / Saul Greenberg



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?

Evan Golub / Ben Bederson / Saul Greenberg

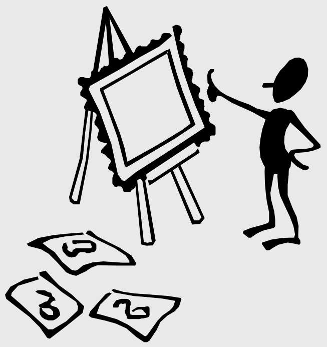
Low fidelity prototypes

Paper-based prototypes

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

Purpose

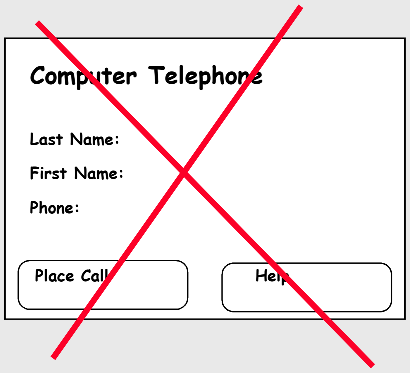
- brainstorm competing representations
- elicit user reactions
- elicit user modifications / suggestions



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

<u>Generally not good to have too much typed!</u> **Should really be hand-drawn on paper.**

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.

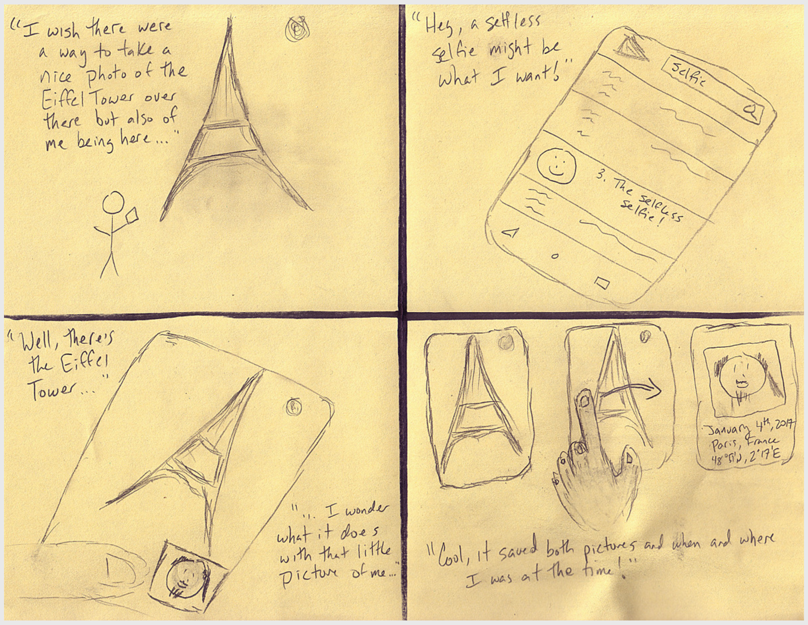
Low fidelity prototypes: Storyboarding

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!

Storyboard of using a new type of selfie app


"I wish there were a way to take a nice photo of the Eiffel Tower over there but also of me being here..."
"Hey, a selfless selfie might be what I want!"
Selfie
3. The selfless selfie!
"Well, there's the Eiffel Tower..."
"... I wonder what it does with that little picture of me..."
January 4th, 2017
Paris, France
48°51'N, 2°17'E
"Cool, it saved both pictures and when and where I was at the time!"

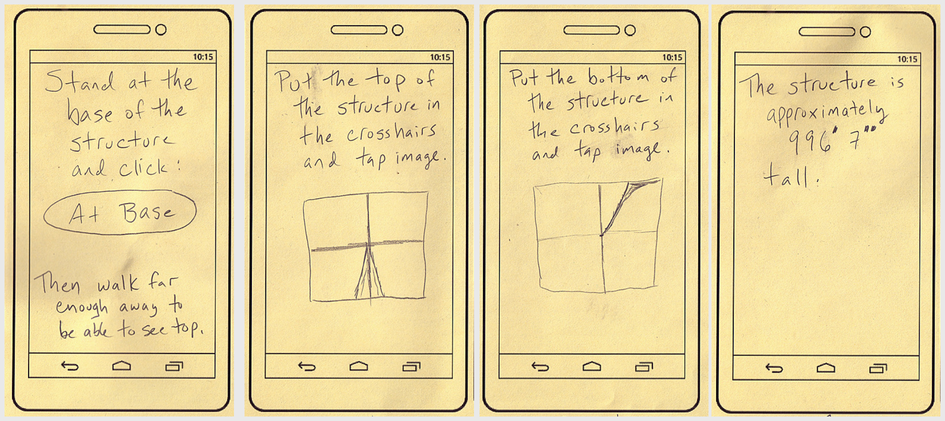
Storyboard of an app to measure height of a structure

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.

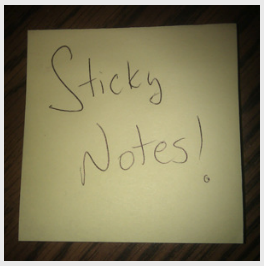
The app uses math and science to calculate the approximate height of the structure! We can use trigonometry on the angles and GPS-based distance…


10:15
Stand at the base of the structure and click:
At Base
Then walk far enough away to be able to see top.
10:15
Put the top of the structure in the crosshairs and tap image.
10:15
Put the bottom of the structure in the crosshairs and tap image.
10:15
The structure is approximately 996' 7" tall.

PICTIVE prototypes

"**P**lastic **I**nterface for **C**ollaborative **T**echnology **I**nitiatives through **V**ideo **E**xploration" - Muller, CHI 1991

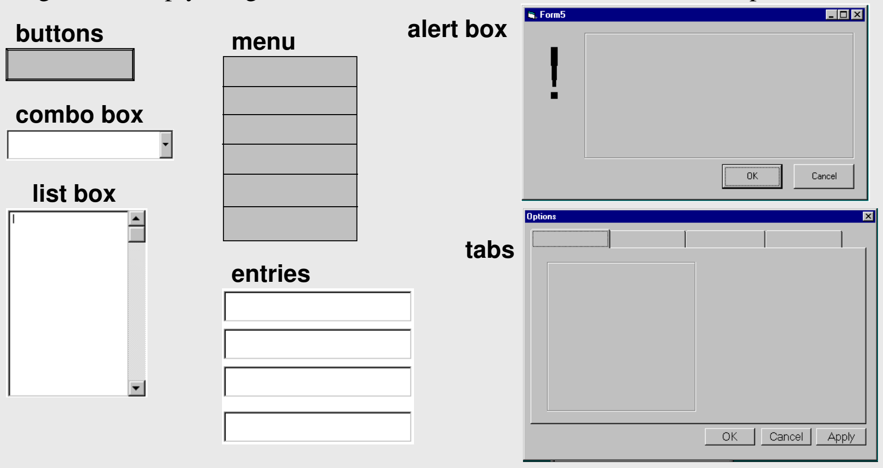
- Design is multiple layers of sticky notes and plastic overlays
  - different sized stickies represent icons, menus, windows etc.
- Interaction demonstrated by manipulating notes
  - contents changed quickly by user/designer with pen and note repositioning
- Session can even be recorded for later analysis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



PICTIVE prototypes

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

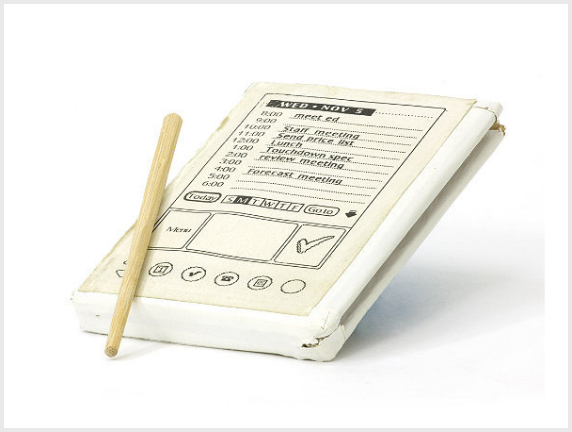
e.g., these empty widgets were created in Visual Basic and could be printed out:



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

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

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.

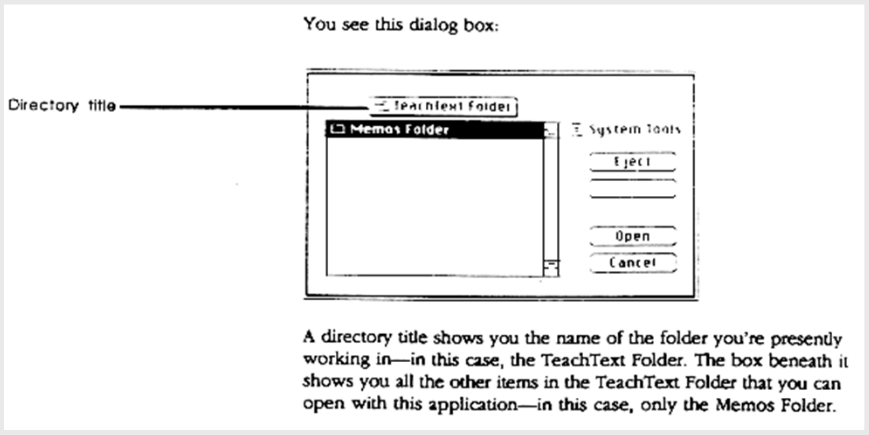
Other uses of low fidelity prototypes

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…

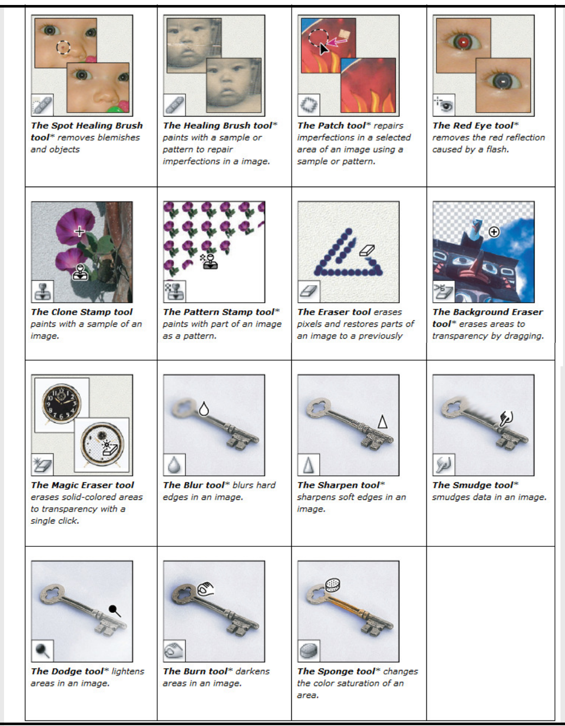
Evan Golub / Ben Bederson / Saul Greenberg


You see this dialog box:

Directory title
TeachText Folder
Memos Folder
System Tools
Eject
Open
Cancel

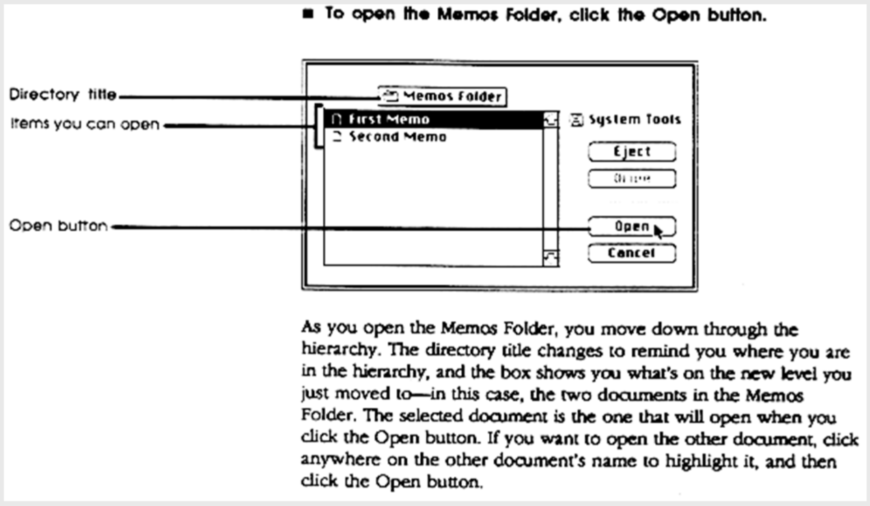
A directory title shows you the name of the folder you're presently working in—in this case, the TeachText Folder. The box beneath it shows you all the other items in the TeachText Folder that you can open with this application—in this case, only the Memos Folder.

1908s - From Apple's Tutorial Guide to the Macintosh Finder



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

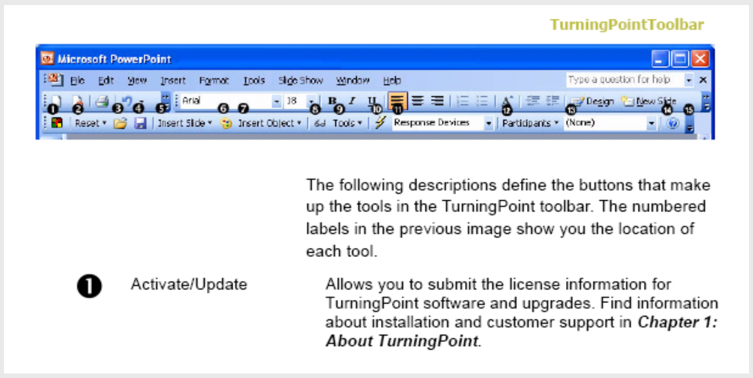
Photoshop CS2
Retouching tools gallery from help


Directory title
Items you can open
Open button
Memos Folder
First Memo
Second Memo
System Tools
Eject
Drive
Open
Cancel

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

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
Microsoft PowerPoint
File Edit View Insert Format Tools Slide Show Window Help
Type a question for help
Arial 18
Design New Slide
Reset Insert Slide Insert Object Tools Response Devices Participants (None)

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.

❶	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

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=-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

Some tools…

Software that allows you to prototype other software includes PowerPoint, InVision, MarvelApp, Moqups, Balsamiq, 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!

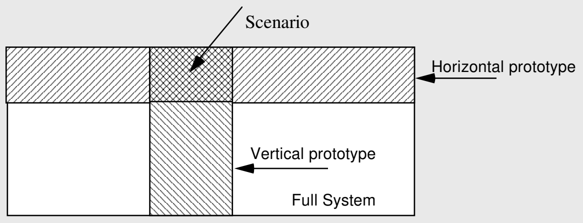
Management may think its real!!!

Evan Golub / Ben Bederson / Saul Greenberg

Medium fidelity prototypes

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



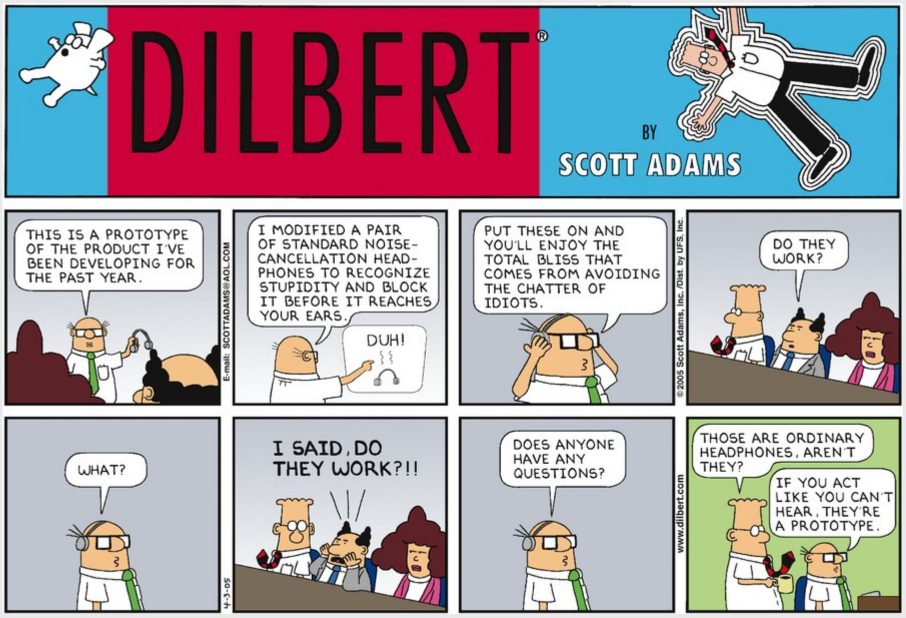
Evan Golub / Ben Bederson / Saul Greenberg

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
BY SCOTT ADAMS
THIS IS A PROTOTYPE OF THE PRODUCT I'VE BEEN DEVELOPING FOR THE PAST YEAR.
I MODIFIED A PAIR OF STANDARD NOISE-CANCELLATION HEAD-PHONES TO RECOGNIZE STUPIDITY AND BLOCK IT BEFORE IT REACHES YOUR EARS.
DUH!
PUT THESE ON AND YOU'LL ENJOY THE TOTAL BLISS THAT COMES FROM AVOIDING THE CHATTER OF IDIOTS.
DO THEY WORK?
WHAT?
I SAID, DO THEY WORK?!!
DOES ANYONE HAVE ANY QUESTIONS?
THOSE ARE ORDINARY HEADPHONES, AREN'T THEY?
IF YOU ACT LIKE YOU CAN'T HEAR, THEY'RE A PROTOTYPE.

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

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

Reading

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