CS3205 – HCI in Software Development

More Prototyping Strategies

Tom Horton

*** Material from:**
Floryan (UVa)
Klemmer (UCSD, was at Stanford)

What Are We Doing Today?

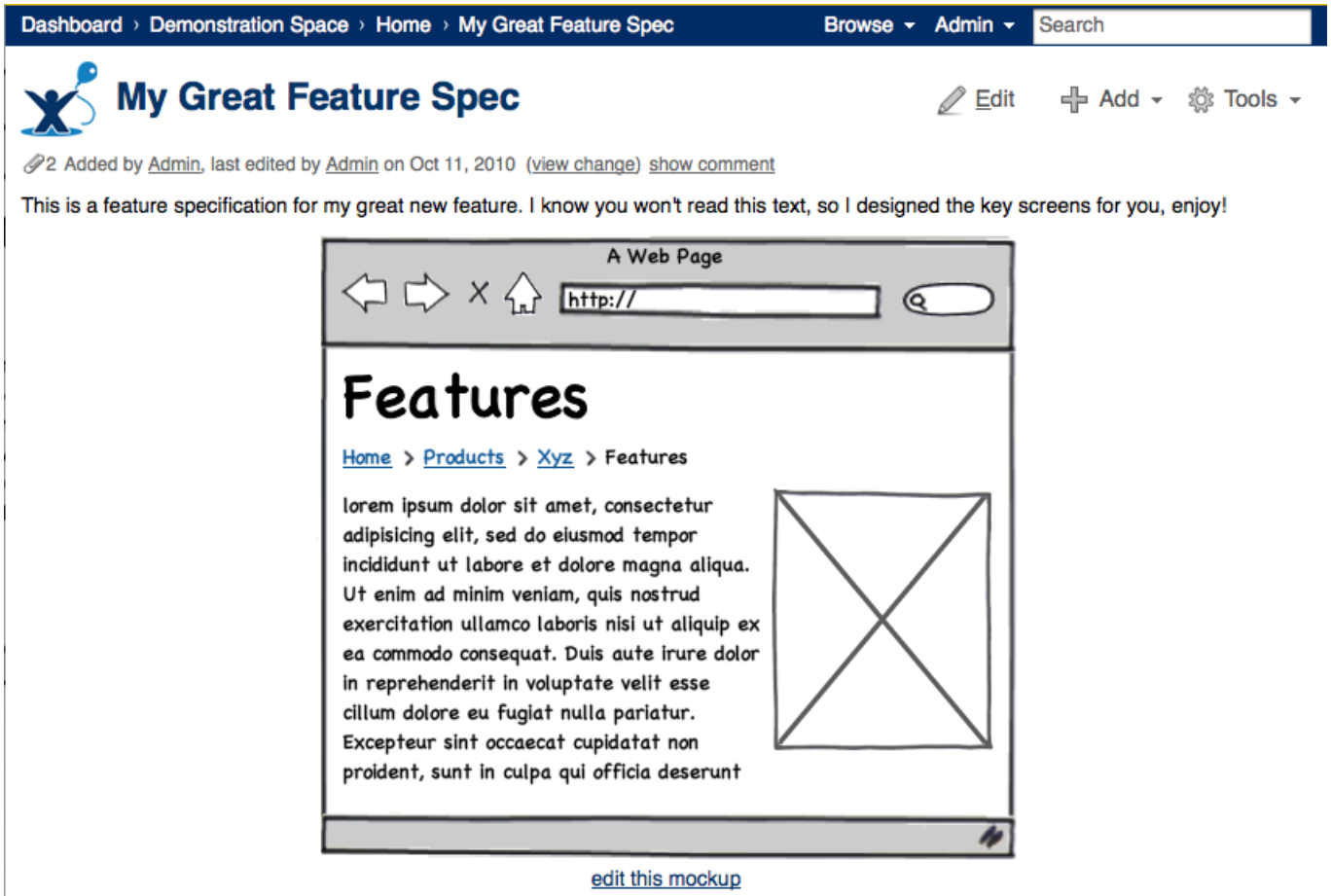
- High-fidelity, Mock-ups

- Wizard of Oz Prototyping

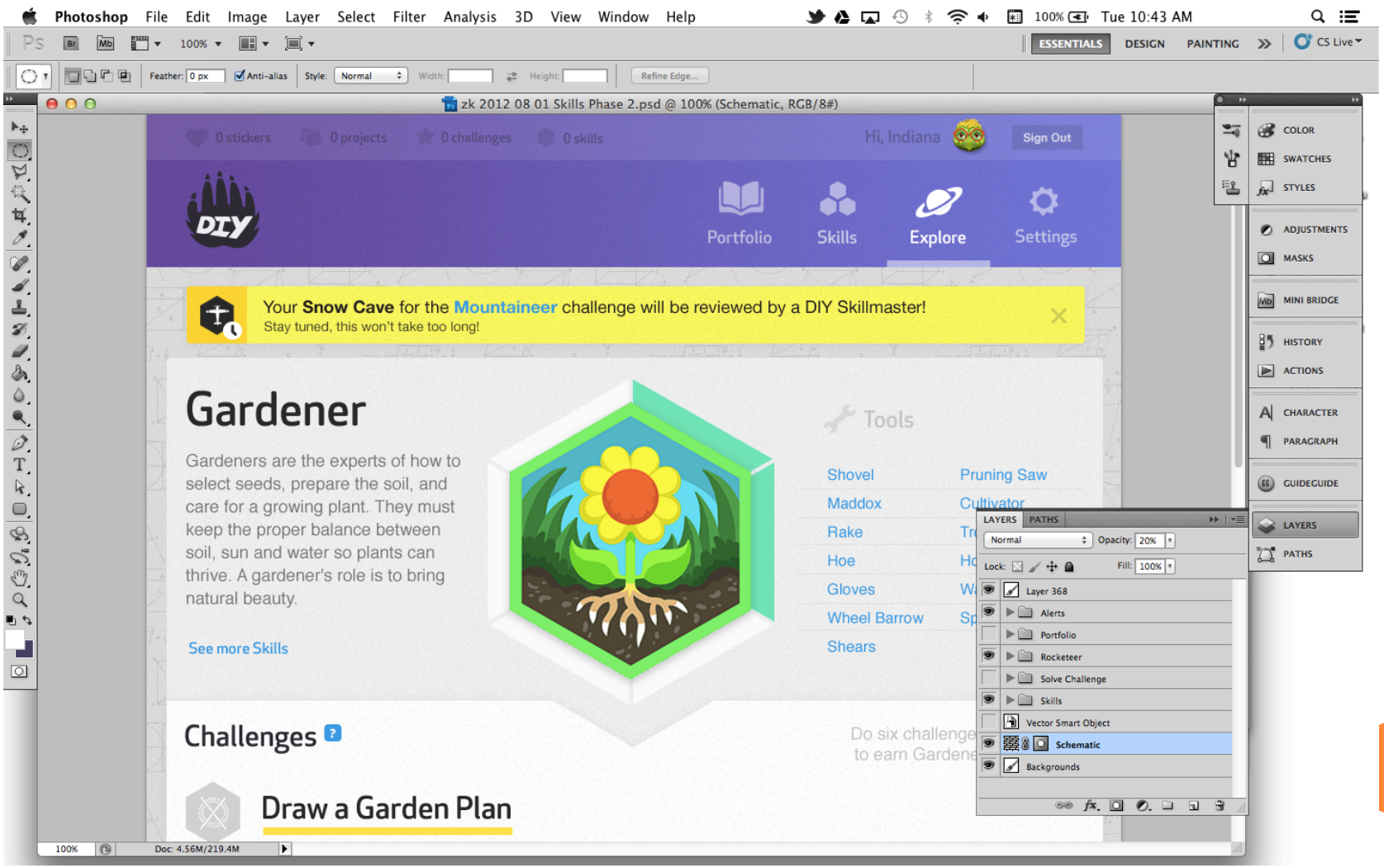
What is a High-Fidelity Prototype?

- "A prototype that mimics the design very closely, but remains functionally incomplete."
- This might be actual code, or actual physical devices that don't work fully.
- For things on screens, how might this be created?
  - Using the final, target implementation language
    - Qt, Java Swing, Tkinter, HTML, Android,…
    - How's that different than version 1?
  - Using a tool to create a wire-frame or mock-up
    - How's that different than low-fidelity prototyping?

Wire-Frames, Digital Mockups



Digital Mockups



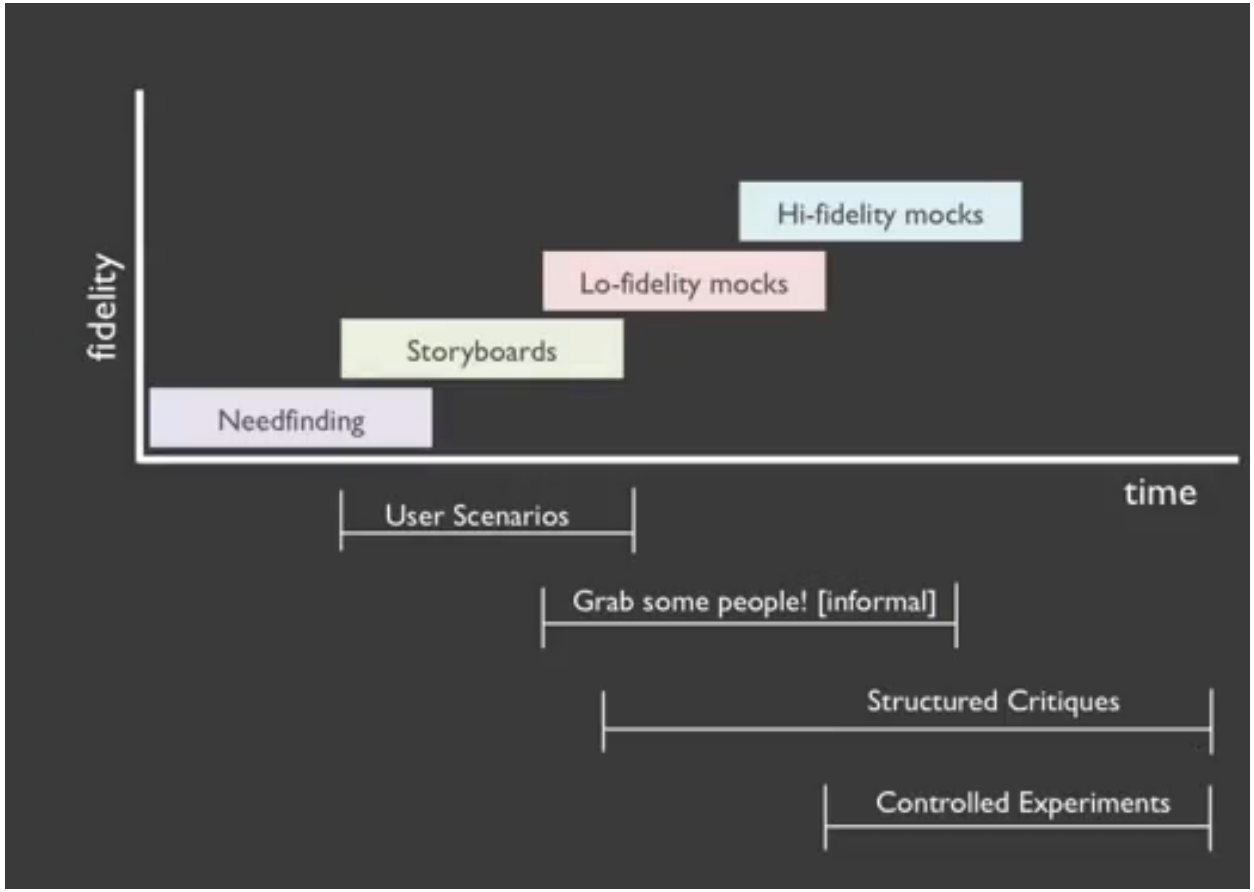
Observations about Examples

- Both clearly are not “functional”
- Balsamiq example:
  - Looks low-fi. Does it really mimic the final design closely? What’s missing?
- Photoshop
  - What’s there?
    - Layout, organizations, size of things.
    - Color, fonts
- Which took more time to create?

Tools

- Free software exists for doing this!
  - http://mashable.com/2012/06/07/mockup-tools/
- Balsamiq Mockups (desktop application)
- Mockingbird (online in-browser mockups)
- Mock up Builder
- POP (Prototyping on paper) app
- Can also use tools like Powerpoint or Photoshop to simulate a design relatively effectively.

Prototyping Over Time



When Does it Become Hi-Fidelity?

- When you:
  - take the time to put in more and more detail,…
  - that's closer to the final design, …
  - that includes "fit and finish"
- When the goal is
  - More about refinement about a design you've chosen,
  - And less about exploring alternatives.

- You might be using the same tool to create the representation, so that's not necessarily the difference.

Wizard of Oz Prototypes

Motivation: Prototyping Problem

- Need feedback from real people in order to improve your design. However,
- Can't get feedback unless you have something built and working!
- What if your system is very complicated?

Motivation: What if we could…

- Make an ***interactive*** application without (much) code, but still:

- Get feedback from real users.

Wizard of Oz Prototyping!

Wizard Of Oz Prototyping

- Human operator simulates the functionality of the system behind the scenes.

- Don't need to build the whole system! Just have your "wizard" simulate it for the user.

Wizard Of Oz Technique

- Make an interactive application without much code.
  - Front end interface is coded.
  - A remote wizard controls the interface's characteristics based on user input directly.
  - Makes sense to do this if it's faster/cheaper/easier than making a real thing.
- Get feedback from real users.
  - Users will think the system is more real if done well!

Another Example: Speech Recognition

- There was a time when:
  - Speech recognition software was NOT good at all (still not great).
  - We knew it would get better.
  - We wanted to know what user's interactions with this technology should be like.

- How to study this without building a speech recognition system ourselves?

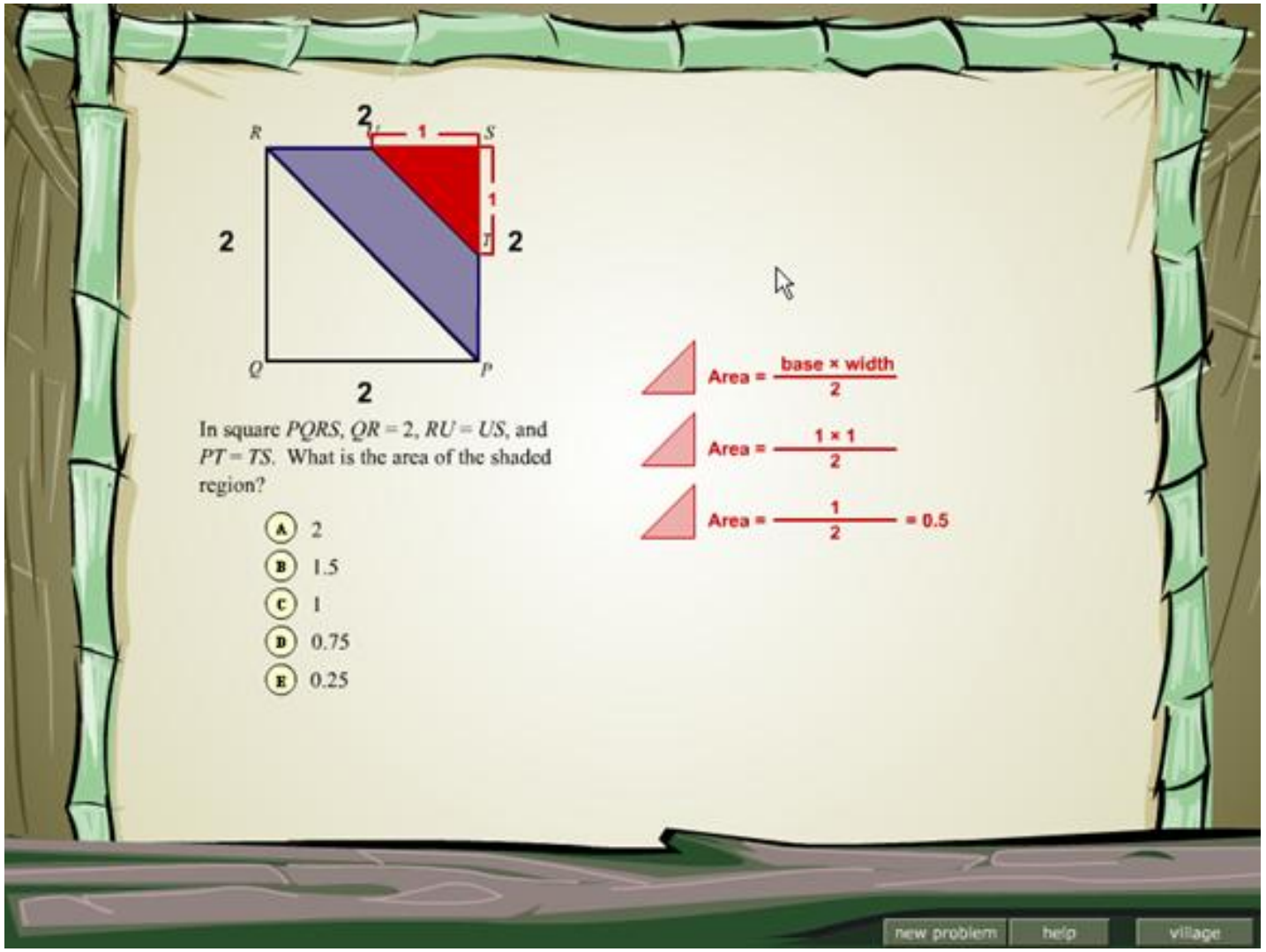
Another Example: Speech Recognition

- Wizard of Oz the interaction!
  - Human listens to the voice commands of the user.
  - Adjusts the interface accordingly.

Funny Example

- https://www.youtube.com/watch?v=AlyvYLLtQOg

Another Example: Intelligent Tutors!



When to use Wizard Of Oz?

- Useful when:
  - There is an advanced technology in your system that you don't have time to build / incorporate into prototype.
    - Speech recognition, artificial intelligence, etc.
  - You haven't determined how best to implement a feature (e.g., personalized feedback) and want to test it first.

Paper Prototypes Are Kinda "Woo-like"

- In low-fidelity evaluations, "playing the computer" is a form of WOO
  - You are essentially the "Wizard" controlling interactions with a paper prototype while users use it.
- However, remember that the more 'real' the wizard's mirage is, the more realistic the user's reactions will be.
- Note: No one would say "paper prototypes are an example of WOO."

Making A WOO Prototype

- 1. Map out scenarios and application flow.
  - Enumerate ALL scenarios if possible.
  - If not, provide guidelines in as specific a format as possible.
- 2. Put together interface “skeletons”
- 3. Develop “hooks” for wizard input.
  - If paper, hooks will be very manual.
  - If done in software, some code will need to be developed.

Making A WOO Prototype

- 4. Put it all together.
  - Where and how will the wizard provide input?
  - How will the wizard receive input from the user?

- * *Remember that you'll need to build actual software for wizard's role eventually, so it must be possible!*

- 5. Rehearse wizard role with a colleague.
  - Being the wizard is surprisingly difficult.
  - Work out easy bugs in interaction before using real users.

Collecting Data for WOO Prototypes

- Practice with friends first.
- Once comfortable, recruit “users”
- Two roles: facilitator and wizard
  - Facilitator: Provides tasks (paper) and takes notes.
  - Wizard: Operates the interface.

Collecting Data for WOO Prototypes

- User feedback can be:
  - Think aloud (speak freely as performing tasks)
  - Retrospective (discuss task afterwards)
  - Heuristic Evaluation (experts are watching interaction unfold)
- Interaction is normally video taped so designers can review later.
- Always debrief users, reveal the wizard if necessary / appropriate.

Wizards Throughout Development

- Functionality vs. Time graph

Advantages of Wizards

- Faster to make / cheaper, thus more iterative prototypes possible.
- Creating multiple variations is very easy (no code to rewrite).
- More “real” than pure paper prototyping or mockups.
- Identifies bugs and problems with current design.
- Places user at center of development.
- Can envision challenging to build application.
- Designers learn by playing wizard.

Disadvantages of Wizards

- Can you really "hide behind the curtain" in a non-distracting way?
- Simulations may represent otherwise imperfect (or impossible) tech.
- Wizards require training and can be inconsistent.
- Playing the wizard can be exhausting ☹
- Some features are difficult (or impossible) to simulate perfectly.
- May be inappropriate in some venues.

Prototypes: Quantity vs. Quality

- Is it better to produce a large quantity of designs, or to focus on creating the best one design?

Quantity vs. Quality

- Bayles and Orland put this to the test.

Quantity vs. Quality

- ***Well, come grading time and a curious fact emerged: the works of highest quality were all produced by the group being graded for quantity. It seems that while the "quantity" group was busily churning out piles of work - and learning from their mistakes -- the "quality" group had sat theorizing about perfection, and in the end had little more to show for their efforts than grandiose theories and a pile of dead clay***. [Bayles, Orland]

Functional Fixation

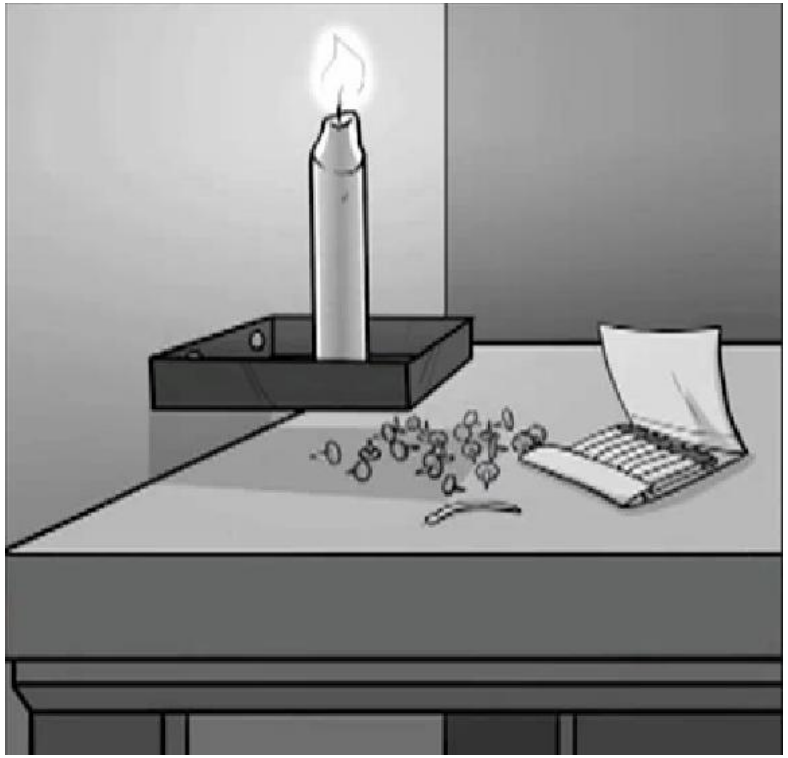
- You make a set of decisions that produce one prototype (or anything)
- You get feedback or (re)evaluate it
- But the improvement you make to this are minimal
  - You stick with the same essential design approach
  - You are reluctant to break the mold and try something very different

- You get an idea and stick with it! (It's human nature! Search for psychological concept of "functional fixedness.")
  - https://en.wikipedia.org/wiki/Functional_fixedness
  - https://www.mendix.com/blog/functional-fixation-and-the-power-of-parallel-prototyping/

Functional Fixation

[Duncker, 1945]

Functional Fixation



[Duncker, 1945]

Better Approach?

- Parallel Prototyping:
  - Making multiple prototypes in parallel
  - Studies show that this leads to better designs!
    - i.e., designs led to statistically higher values in quantifiable variables of interest (more on this later).
    - *Klemmer, Gentner, Loewenstein, Thomson, etc.
- Separates Ego from Artifact
  - i.e., a criticism of one design is NOT a criticism towards the designer.
- Supports TRANSFER of positive attributes across designs.

Ok fine…so how do we compare prototypes?

- We perform an evaluation!
- An ***evaluation*** is an experiment (or set of experiments) meant to provide answers to at least one design question.
- The next topic!
- https://www.youtube.com/watch?v=kCSzjExvbTQ

Questions?