CS3205 – HCI in Software Development

Introduction To Prototyping

Tom Horton

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

Readings

- ID Book. Chapter 11 in published book, “Design, Prototyping, and Construction.” (Ch. 10 in our eBook)
- Other readings and videos on the web

What is a prototype?

- What do you think of when you hear “prototype”?

- What kinds of prototypes have you seen anywhere?
  - in other fields or disciplines?
  - on television?

- What are they “for”?

What is a prototype?

- In other design fields a prototype is a small-scale model:

  a miniature car
  a miniature building or town

- Exists for some purpose
  - Show the "concept" to some stakeholders
  - Get feedback about some aspect
    - Perhaps compare two approaches
  - Test something in an experimental way
    - E.g. a wing in a wind-tunnel

Prototyping and Software

- Do software companies do this?
  - Sometimes do it well
  - But sometimes the prototype is…

Version 1.0!

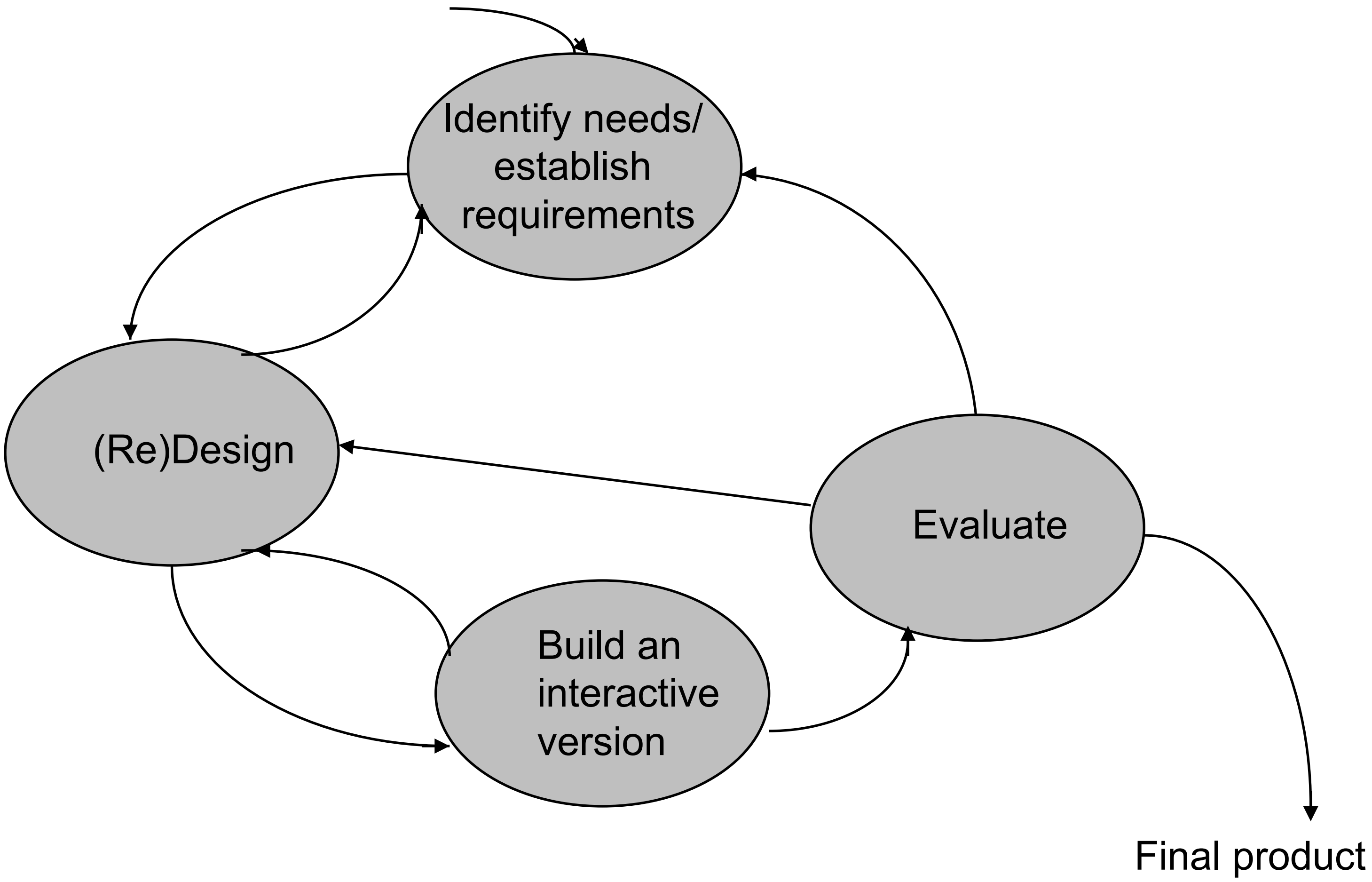
- Constantine and Lockwood:

  "Software is the only engineering field that throws together prototypes and then attempts to sell them as delivered goods."

What is Prototyping?

- Prototyping is:
  - “The rapid creation of an approximation to a design idea for the purpose of retrieving feedback and knowledge.”

Remember? A model for interaction design

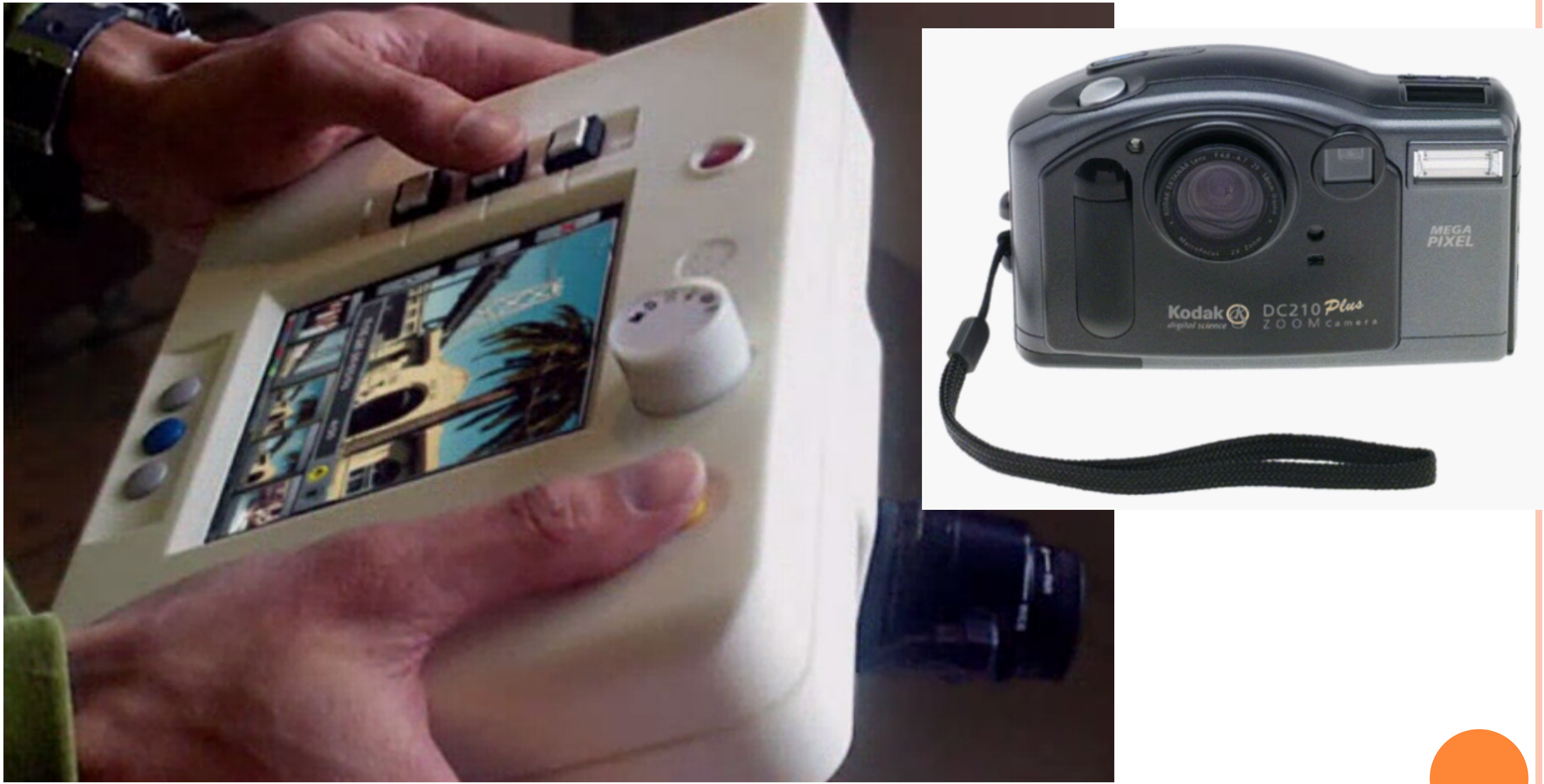

Identify needs/ establish requirements
(Re)Design
Evaluate
Build an interactive version
Final product

What is Prototyping?

- We prototype to:

1. Gain insights into user behavior.
2. Communicate ideas to other teammates and stakeholders.
3. Collect data for arguing the best design choice.
4. Prototypes encourage reflection, a very important aspect of any kind of design activity.

EXAMPLE: IDEO DIGITAL CAMERA



No Computation on device!
No Pictures (No lens, etc.)

Lessons from IDEO

- Prototypes are nearly ALWAYS incomplete.
- Goal is to SIMULATE specific aspects of the design and acquire knowledge regarding these targeted aspects.
- IDEO wanted to know more about the digital aspects of the camera.

When and at What Level

- You might prototype at various times in the stages of you product's development
  - Different goals, different techniques

1. Conceptual Design
2. Interaction Design
3. Screen Design

(1) Prototyping: Conceptual Design

- Early in development
- Explore high-level issues
  - Different conceptual models
  - Interaction styles
  - User needs and characteristics
  - Usability goals

- High-level representations
  - Far from final code or GUIs

(2) Prototyping: Interaction Design

- Later in development
- Focus on user work-flows
  - Tasks and scenarios you’ve identified
- Might focus at the screen (or page) level. Possibly like this:
  - Identify screens, pages, activities
  - Organize these in groups
  - Define flows or transitions between them
- Involve users in evaluation

- Representations: Still probably not much like final code or GUIs
  - Must be able to be “interactive” somehow

(3) Prototyping: Screen Design

- Before development
- Define and refine screens (pages)
  - Blue-prints for final physical design
- User evaluation
  - Both achieving tasks and navigation, and other usability criteria (as we've studied)

- Representations: Something clearly close to final implementation of screens and pages
  - Perhaps just showing form or layout (not interaction)

Prototyping Can Help You Find…

- ***‘Known Unknowns’***: Aspects of a design that you know you don’t understand and wish to learn.
  - i.e., “Which color scheme is most user friendly?”
  - i.e., “Should this be a mobile or desktop application?”

- ***‘Unknown Unknowns’***: Aspects of a design that you don’t know are open issues.
  - i.e., “Why can’t old people use this tiny screen!?”

Prototyping Process

- Prototyping is NOT a process in which you haphazardly create designs that you believe are various levels of “awesome”.

- Prototyping is about defining questions regarding your designs, and building something that answers those questions.

Prototyping Process

1. What are your goals for the prototype? What do you wish to learn?

2. How can you measure whether or not that goal has been achieved? How can you measure which of multiple prototypes is superior?

3. What is the MINIMUM amount of work necessary to produce, measure, and learn from your prototype?

1. Prototyping Goals

- If you perform your user and task analysis well, then these are represented by your **usability goals and requirements**.
- Thus we can select the usability requirements (probably a subset) that we wish to test with a given prototype.

1. Prototyping Goals

- Example:
- **UR**: *User must be able to type text messages at a minimum of 50 wpm while making less than 3 errors on average.*

- What do you wish to learn?
  - What physical interface is best for supporting users that wish to text and type on their phone?
  - Prototype choices:
    - Physical keypad
    - Touchscreen
    - Others?

2. Measuring Goals

- We can think about the *falsifiability* of usability requirements.

- If requirements are truly falsifiable, then we should be able to design user studies that prove (based on data) that our prototype is sufficient.

2. Measuring Goals

- Example:
- **UR**: *User must be able to type text messages at a maximum of 50 wpm while making less than 3 errors on average.*

- How to measure this:
  - Pretty simple, give users different phone prototypes and see how well they type.
  - In reality, a bit more complicated than this, but a good start.

3. Minimum Work Necessary

- What is the absolute minimal working prototype with which we can perform our user study?

- Strip out every feature that is not absolutely necessary.

3. Minimum Work Necessary

- Example:
- **UR**: *User must be able to type text messages at a maximum of 50 wpm while making less than 3 errors on average.*

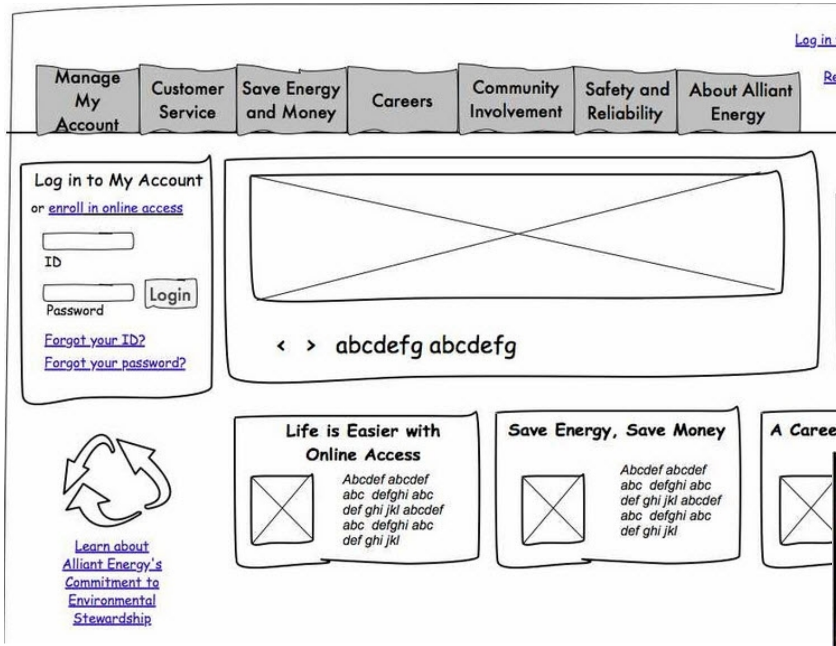
- Minimal:
  - Physical devices that can be typed on, need to be able to record speed and errors while typing.
- Unnecessary features:
  - Sending texts to other phones
  - Selecting contacts to send to
  - Etc.

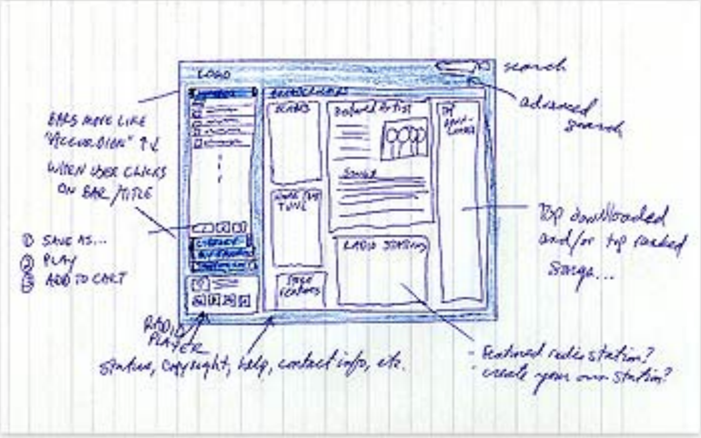
3. Minimum Work Necessary

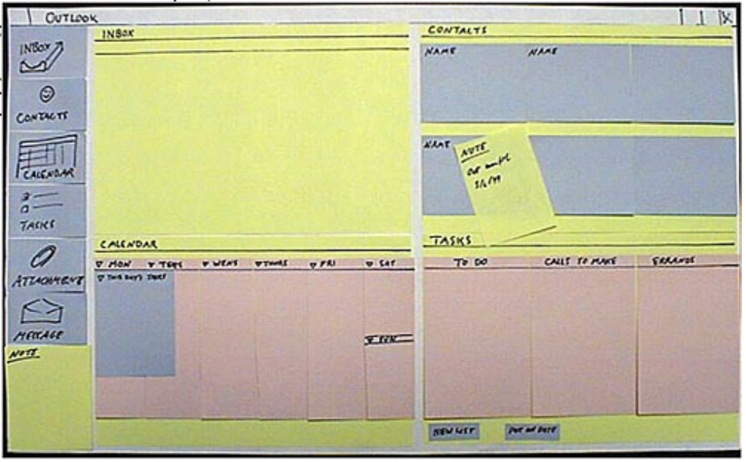
- Show graph on board (Time Spent vs. Learned)!

Types of Prototypes

- Low-Fidelity:

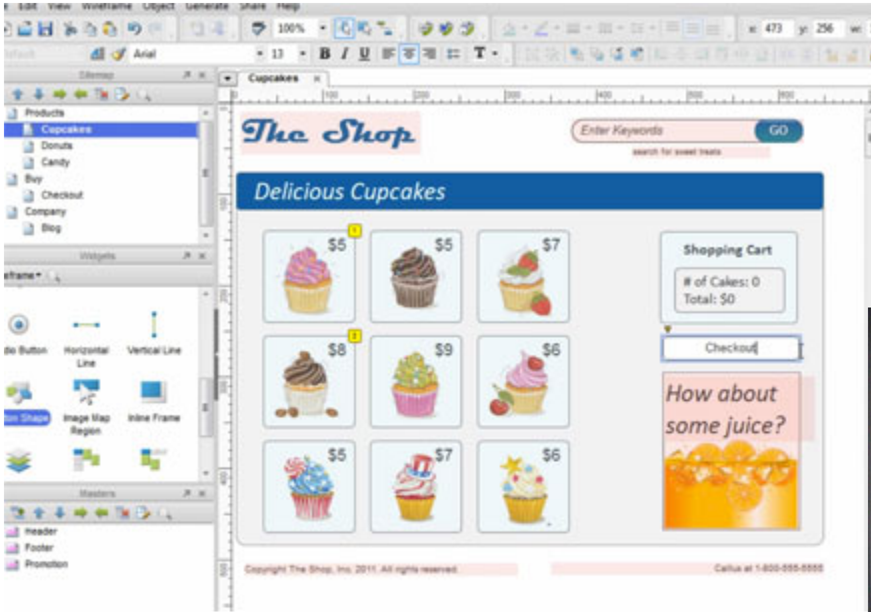


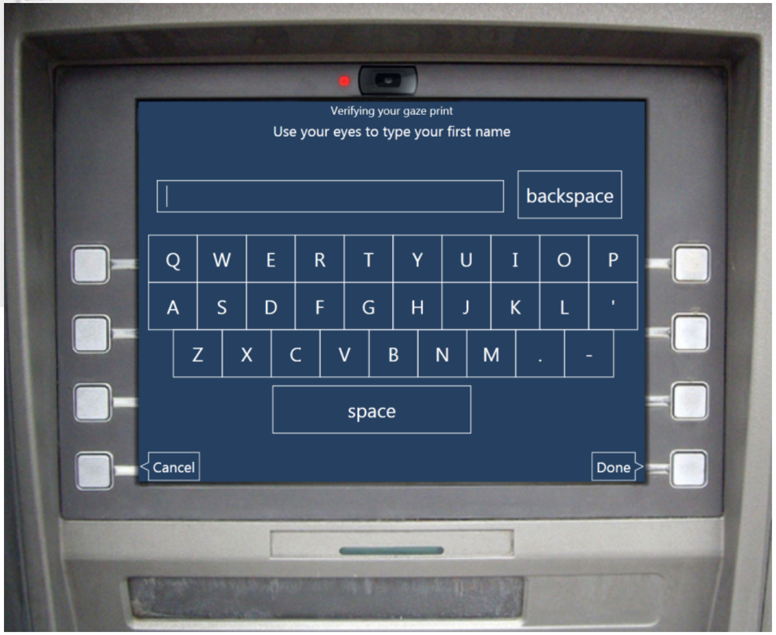

LOGO
search
advanced search
BARS MOVE LIKE "ACCORDION" ↑↓ WHEN USER CLICKS ON BAR/TITLE
① SAVE AS...
② PLAY
③ ADD TO CART
Featured Artist
Songs
RADIO STATIONS
Top downloaded and/or top ranked songs...
RADIO PLAYER
Status, copyright, help, contact info, etc.
- Featured radio station?
- create your own station?



Types of Prototypes

- High-Fidelity:




Verifying your gaze print

Use your eyes to type your first name

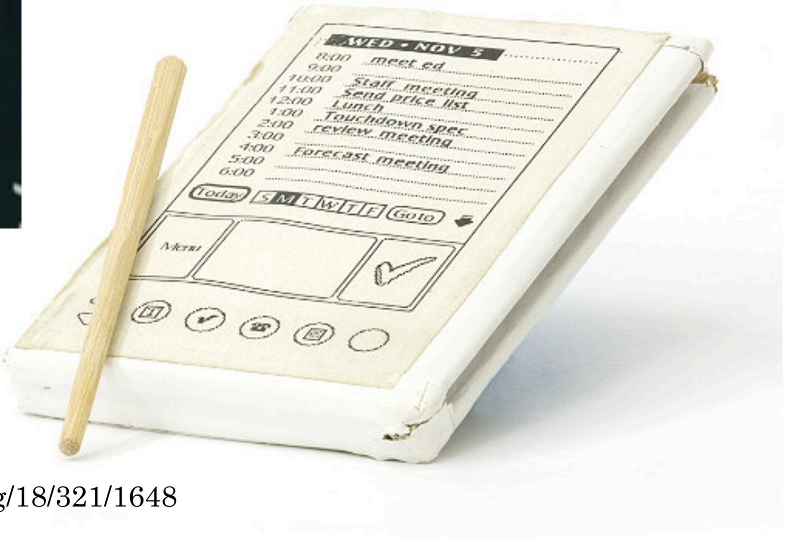
Prototyping Sets Over Time

- Graph on board (num prototypes vs time)

A Lot of Prototypes is GOOD!

“If you want to have good ideas you must have many ideas. Most of them will be wrong, and what you have to learn is which ones to throw away.” -- Linus Pauling

Jeff Hawkins' Block Of Wood (1995)



http://www.computerhistory.org/revolution/mobile-computing/18/321/1648

Jeff Hawkins' Block Of Wood

- Did NOT learn about:
  - Battery Life
  - Interface and other digital interactions
  - Sound effects, feedback, etc.

- DID learn about:
  - Form Factor
  - Living with the device 24/7

Types of Prototypes:

- ***Feel***
  - What does it look and feel like? What's the form?
  - Note: doesn't really need to "work" to achieve this.

- ***Implementation***
  - How does it work? How does the user interact with it? What are the steps, the flow, the feedback, etc.?
  - Note: some aspects of the prototype must be interactive, even if in some limited way.

- ***Role***
  - What is the experience like for the users?
  - When will they use it? In what context?

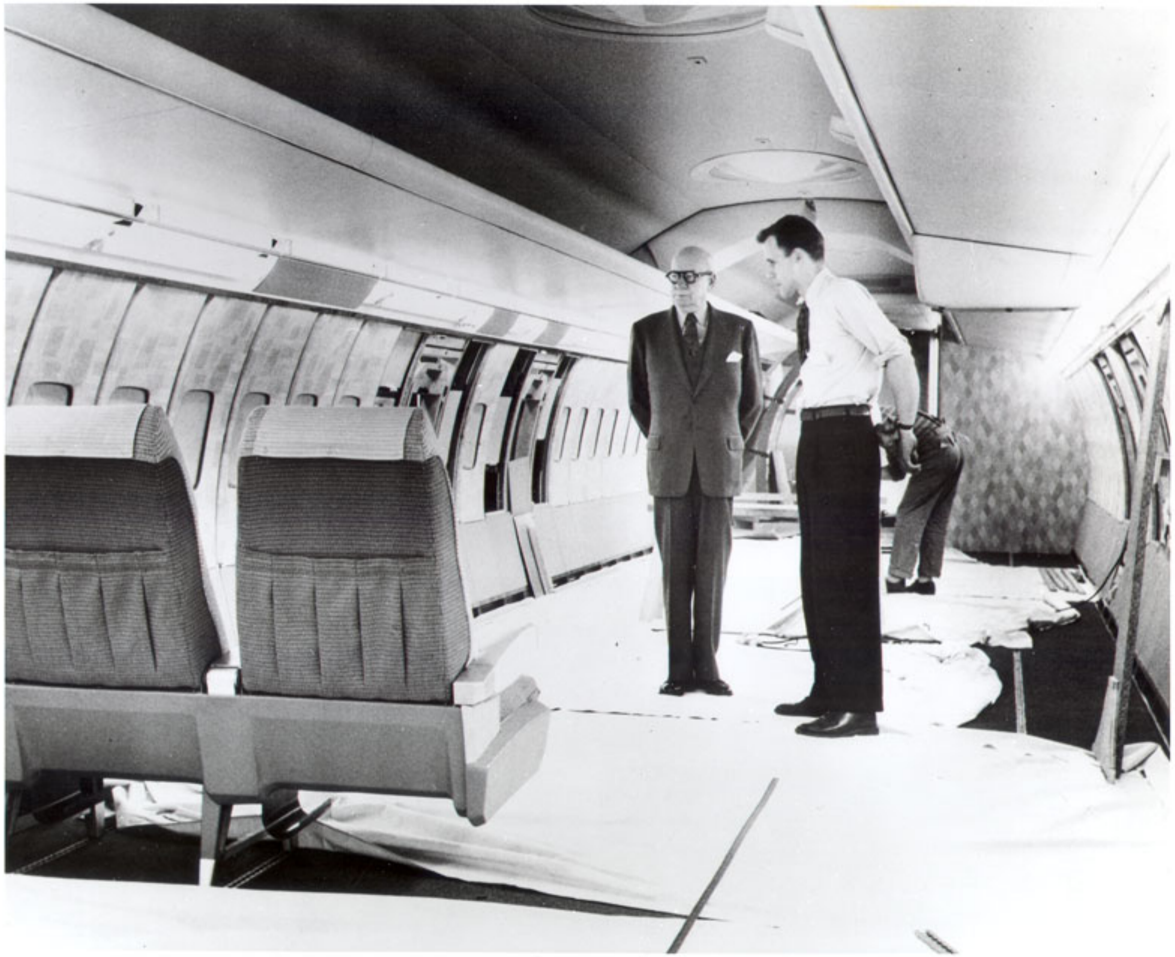
Prototyping Rules

- Prototypes should NOT be required to be complete.
- Prototypes should NOT need to be updated.
- Prototypes should be easy to change.
- Prototypes should be disposable.
  - Or, at least, you should mentally be prepared to dispose them.

- In prototyping, the goal is NOT the artifact itself!
  - It's what you learn from the **feedback** you get from building, evaluating, and iterating on them.

Prototypes Can be BIG!

- Walter Teague. Mockup of Boeing 707 airline cabin, 1950s.



Another BIG Prototype

Cost of Change Over Time

- Prototypes are MORE expensive to change the more high-fidelity they become.
  - SO BE CAREFUL!

- The most common estimate is that it's **100 times cheaper** to make a change before any code has been written than it is to wait until after the implementation is complete [Jakob Nielsen].

- Twenty years of usability engineering experience uniformly indicates that the **biggest improvements in user experience come from gathering usability data as early as possible** in a design project [Jakob Nielsen].

Prototyping Strategies

- A few we will look at:
  - Paper Prototyping
  - Digital Mockups
  - Video Prototypes
  - Wizard of Oz Prototypes

PROJECT PART 3!

- Let’s talk about Part 3 of the Project! ☺

Project Part 3: The Basics

- Due Date: March 28, 11:30 pm

- What will you be doing?
  - Building at least one prototype per team member.
  - Short write-up containing a description of the prototypes, your design decisions, images of the prototypes, etc.
  - We'd like to see at least one of the prototypes that's "outside the box".

Building Prototypes

- Must build at least one prototype per team member.

- Must be either a physical or digital prototype that your peers can actually use (again, we'll see techniques to make this easier).

Building Prototypes

- Do I need to code?
  - Not necessarily…depends on what design questions you are trying to answer.

- I strongly suggest that all prototypes be the same ‘type’.
  - Why? In the next homework you will be evaluating and comparing them to one another.

Write-up Should Contain:

- Usability requirements and goals for your prototypes
  - What are you trying to learn from them?
- Description of each prototype
  - Pictures / screenshots of your prototype (I need to see that you actually built something).
  - Don't be lazy! You will need these prototypes in order to do Part 4.
- Description of the important differences.
  - What important design questions are you unsure about, how have you accounted for various possibilities via your different designs?
- Design rational for each prototype
  - What is your conceptual model? What design principles have you applied and why? What else from lecture informed your prototype designs?
- Comparison and analysis of your different designs