CS377E: DESIGNING SOLUTIONS TO GLOBAL GRAND CHALLENGES: *SMART EDUCATION*

# Early Stage (lo-fi) Prototyping

Prof. James A. Landay
Computer Science Department
Stanford University

Autumn 2020
October 5, 2020

1

## Outline

- Selecting tasks
- Storyboarding
- Low-fi prototyping
- Conducting a low-fi test
- Low-fi vs. Medium-fi prototyping



2

***Task.*** The structured set of activities or high-level actions required to achieve a high level user goal.

***what*** a user wants to do



3

## Task-based Design & Evaluation

- Real tasks customers have faced / will face
  - collect any necessary materials
- Do your tasks support the problem you are solving?
- Mixture of simple & complex tasks
  - simple task (common or introductory)
  - moderate task
  - complex task (infrequent or for power customers)

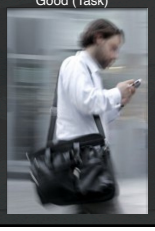


4

## What Should Tasks Look Like?

- Say what customer wants to do, but not how
  - allows comparing different design alternatives

Good (Task)



Tony is visiting London and wants to find the pub that his friend told him about. He is walking down the street using his phone to navigate to the place that he has previously looked up.

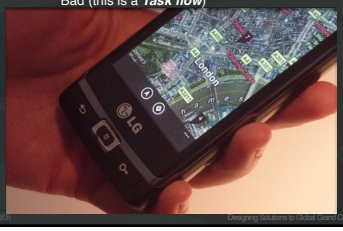


5

## What Should Tasks Look Like?

- Say what customer wants to do, but not how
  - allows comparing different design alternatives

Bad (this is a ***Task flow***)




Tony clicks on the Charing Cross Pub icon and selects "directions to" as he walks down the street.



6

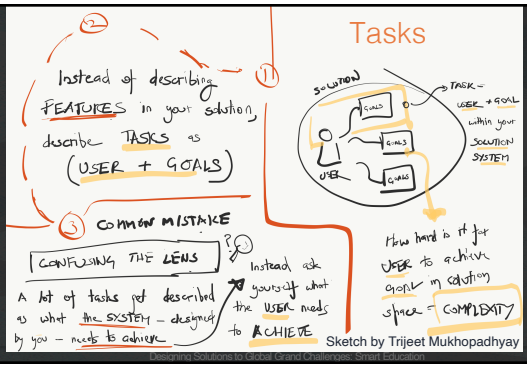
## What Should Tasks Look Like?

- Say what customer wants to do, but not how
  - –allows comparing different design alternatives
- Be specific – stories based on facts!
  - –say who customers are (use POVs or personas or profiles)
    - design can really differ depending on who
    - name names (allows getting more info later)
    - characteristics of customers (job, expertise, etc.)
  - –forces us to fill out description w/ relevant details
- Tasks should usually describe a complete goal
  - –forces us to consider how features work together
    - example: phone-in bank functions



7

## Tasks




Sketch by Trijeet Mukhopadhyay



8

## Using Tasks in Design

- Write up a description of tasks
  - – formally or informally
  - – run by customers and rest of the design team
  - – get more information where needed

**Let my friends know where I am**
Manny is in the city at a club that he wasn't planning to go to and would like to let his girlfriend, Sherry, know where he is and be notified when she is about to get to the club.



9

## Using Tasks in Design (cont.)

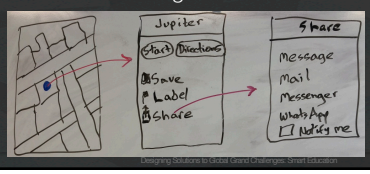
- Rough out an interface design
  - –discard features that don't support your tasks
    - or add a real task that exercises that feature
  - –major screens & functions (not too detailed)
  - –hand sketched
- Produce *task flows* for each task
  - –what customer has to do & what they would see
  - –step-by-step performance of task
  - –illustrate using storyboards (AKA wireframes)
    - sequences of sketches showing screens & transitions



10

## Task Flows Show How to Do the Task

- Task Flows are *design specific*, tasks aren't
- Task Flows force us to
  - –show how various features will work together
  - –settle design arguments by seeing examples
- Show users taskflows to get feedback






11

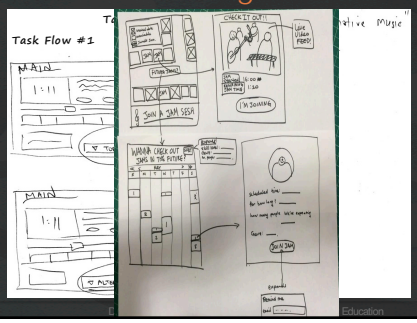
**What** ⟷ **How**

**Task** ⟷ **Task Flow**



12

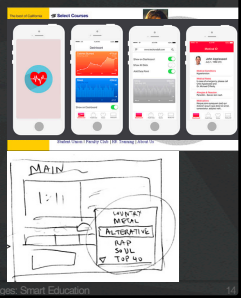
## Storyboards in UX Design




13

## Fidelity in Prototyping

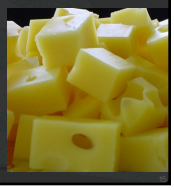
- Fidelity refers to the level of detail
- High fidelity?
  - –prototypes look like the final product
- Low fidelity?
  - –artists renditions with many details missing




14

## Hi-fi Prototypes Warp

- Perceptions of the tester/reviewer
  - –representation communicates "finished"
    - comments focus on color, fonts, & alignment
- Time
  - –encourage precision
    - specifying details takes more time
- Creativity
  - –lose track of the big picture

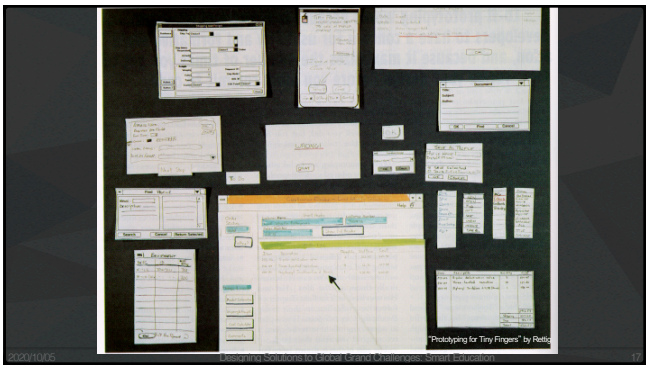


15

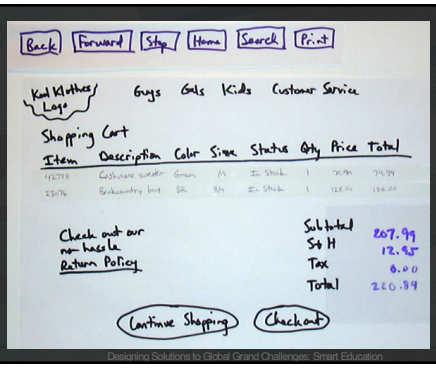
## Why Use Low-fi Prototypes?

- Traditional methods take too long
  - –sketches → prototype → evaluate → iterate
- Can instead simulate the prototype
  - –sketches → evaluate → iterate
  - –sketches act as prototypes
    - designer "plays computer"; others observe & record
- Kindergarten building skills
  - –allows non-programmers to participate

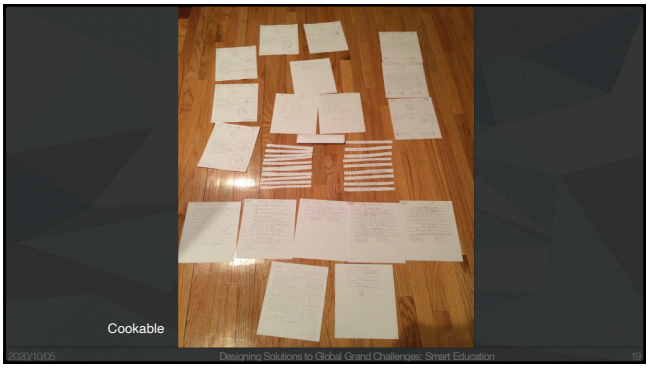
16



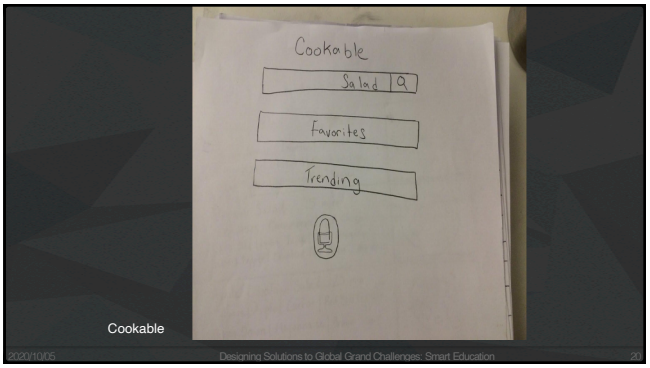

17



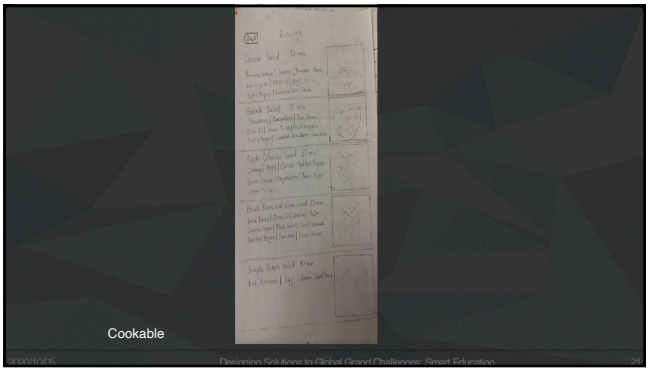

18



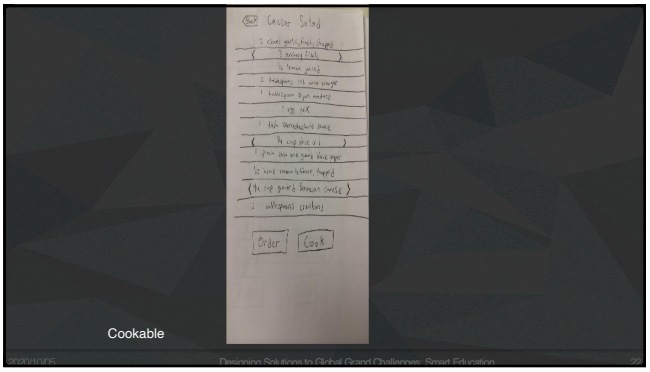

19



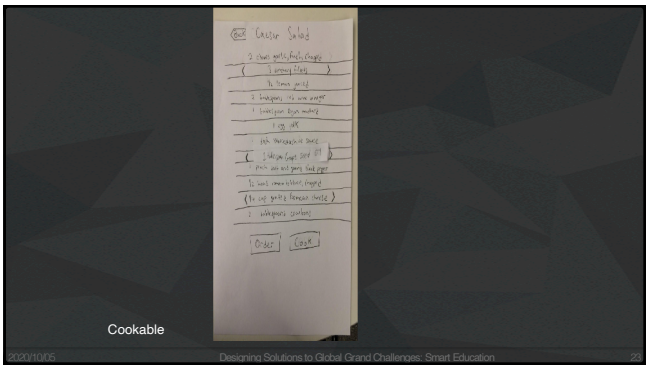

20



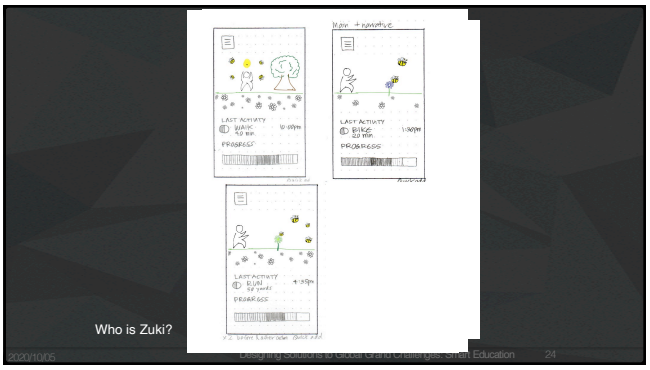

21




22




23

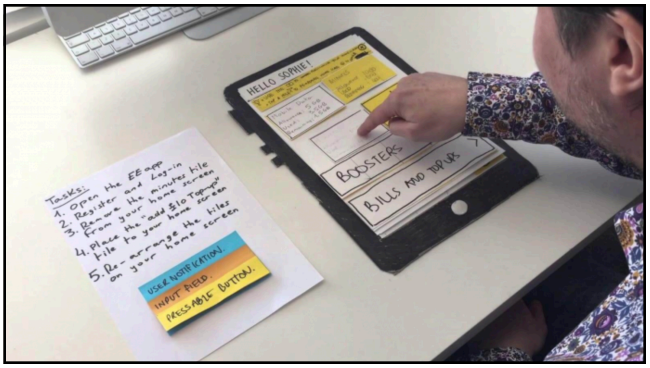



24

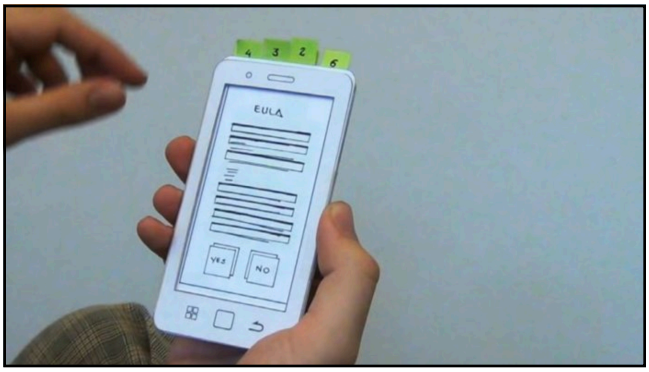
## Constructing the Model

- Set a deadline
  - –don't think too long - *build it!*
- Draw a window frame on large paper
- Put different screen regions on cards
  - –anything that moves, changes, appears/disappears
- Ready response for any user action
  - –e.g., have those pop-up dialogs, etc. already made
- Use printer/scanner to make many versions

25




26




27

## Preparing for a Test

- Select your "customers"
  - –understand background of intended users
  - –use a questionnaire to get the people you need
  - –don't use friends or family
- Prepare scenarios that are
  - –typical of the product during actual use
  - –make prototype support these (small, yet broad)
- Practice to avoid "bugs"

28

## Conducting a Test

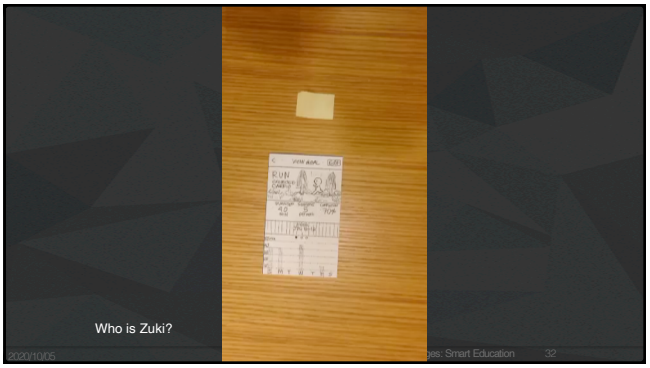
*Four Roles*

- Greeter – puts users at ease & gets data
- Facilitator – only team member who speaks
  - –gives instructions & encourages thoughts, opinions
- Computer – knows application logic & controls it
  - –always simulates the response, w/o explanation
- Observers – take notes & recommendations


29


30


31




32

## Evaluating Results

- High level questions about your design
  - –does it address the problem you want to solve?
  - –is this the right realization of your solution?
- Sort & prioritize observations
  - –what was important?
  - –lots of problems in the same area?
- Make changes & iterate
  - –even iterate between tests



33

## Advantages of Low-fi Prototyping

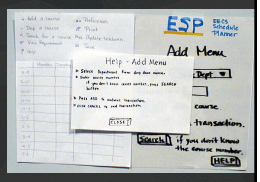
- Takes only a few hours
  - –no expensive equipment needed
- Can test multiple alternatives
  - –fast iterations
    - number of iterations is tied to final quality
- Almost all interaction can be faked (Wizard of Oz)



34

## Problems with Low-fi Prototypes

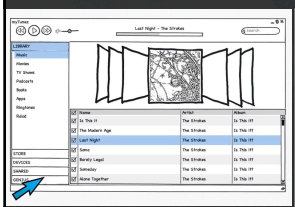
- "Computer" inherently buggy
- Slow compared to real app
  - – timings not accurate
- Hard to implement some functionality
  - – pulldowns, feedback, drag, viz …
- Won't look like final product
  - – sometimes hard to recognize widgets
- End-users can't use by themselves
  - – not in context of user's work environment






35

## Interactive Lo-fi Tools



Balsamiq Mockups
http://balsamiq.com

POP
https://marvelapp.com/pop



36

## Remote Testing of Low-fi Prototypes

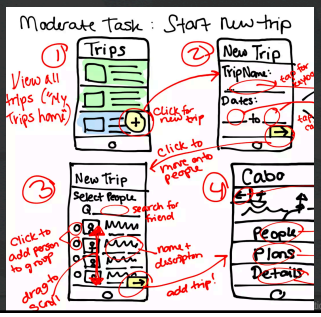
1. Participant runs & records prototype (e.g., Balsamiq/POP) on their phone [hardest]
   - user records interaction by recording screen on iOS/Android
   - you record zoom meeting while participant speaks aloud
   - see https://uxdesign.cc/moderating-ux-research-with-zoom-1d4e88614277
2. Participant runs zoom on their phone while you screen share prototype [moderate]
   - user taps on items & verbalizes aloud
   - ***you control prototype & record meeting***
   - see https://uxdesign.cc/moderating-ux-research-with-zoom-1d4e88614277
3. Participant hugs their laptop [easiest]
   - user runs your prototype (e.g., Balsamiq/POP) on their own phone
   - you record zoom meeting of their screen that is ***captured by their laptop camera***
   - read https://medium.com/@beardicular/were-still-hugging-our-laptops-8e7f22ed8306



37

## Fidelity in Prototyping:

Instagator






38

## Fidelity in Prototyping

Task 1: Take a Destination Poll




39

## Summary

- Prototypes are a concrete representation of a design or final product
- Low-fi testing allows us to quickly iterate
  - –get feedback from users & change right away



40