CSCI 5601

Data Modeling and Prototyping

Evaluating: Cognitive Walkthroughs

CH 9, 12, and CH 20

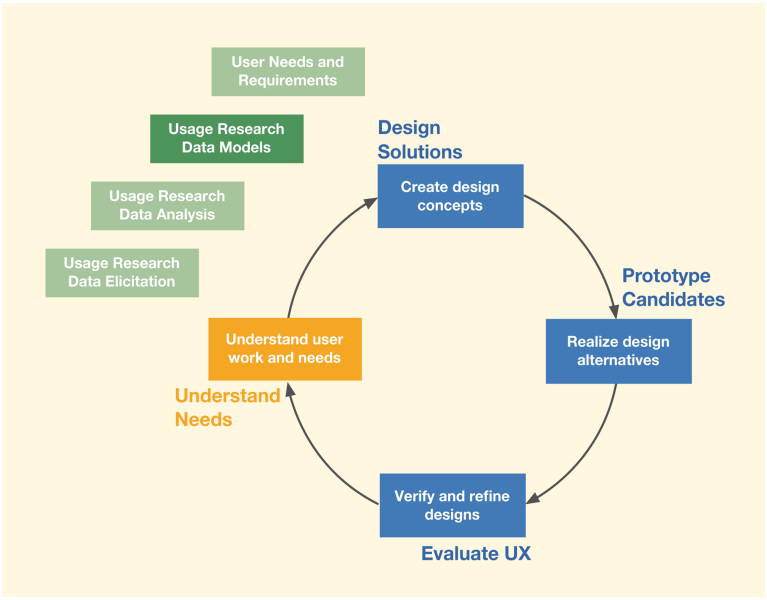
Today

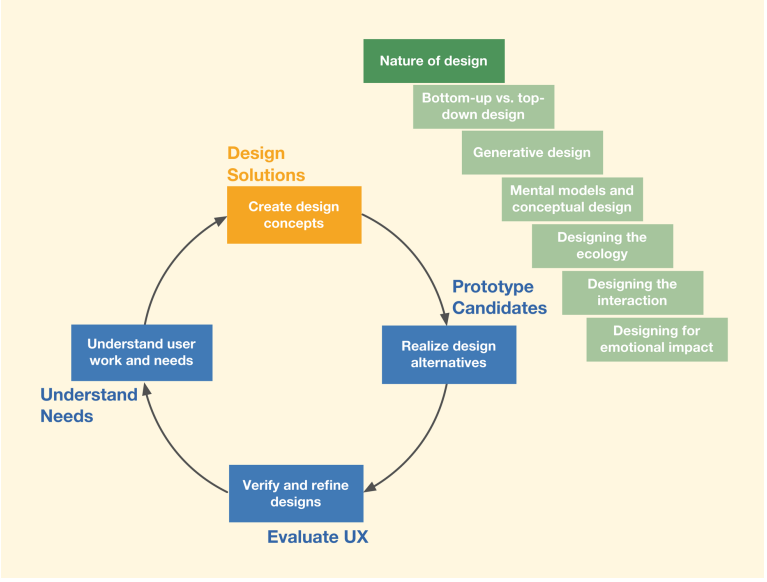
Modeling data and Prototyping

Cognitive walkthroughs

Design: –A2 and the last lab (storyboarding)

Modeling, Prototype and Evaluate: today and next lab





What do we mean by "modeling"?

- "Modeling" sounds very theoretical or academic
- In UX, it's very simple:
  - A way to organize and visualize your usage research data
  - Gives meaning to raw data – it turns usage research data into actionable items for design
  - Later, to inform design

- Uses structured representations to inform UX design
- Each model gives a different perspective into the overall picture of work practice/user needs

Not all models are always needed

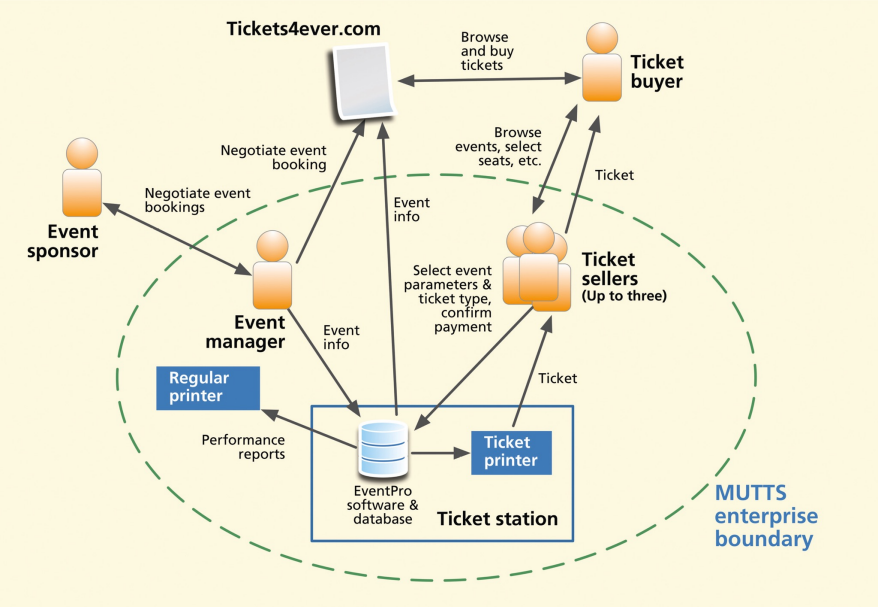
- Most common data models include
  - Flow Models, such as physical, artifact models (seen earlier)
  - **Models for users and task structures**

- Model data helps with design and with communicating to the client about design
  - Such as storyboards – which you did in the lab

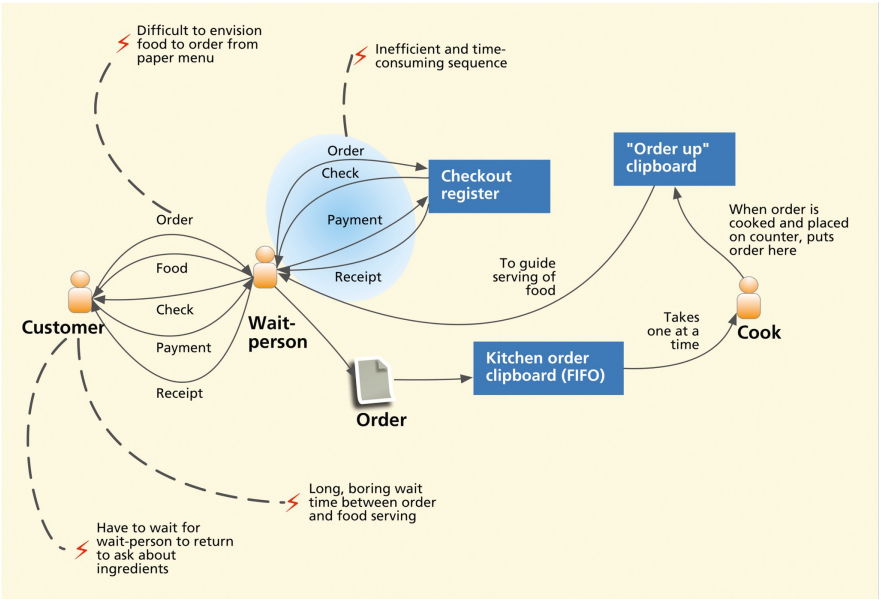
Flow model

- Flow model is a simple graphical representation of how information and artifacts flow through system in the context of work practice
- Example: To show how a piece of music is purchased, downloaded, synced to a device, and played
- How do you do it?
  - Bird's-eye view of entire workflow
  - Nodes for active entities – Work roles, central database
  - Arcs for flow of work, information, non-computer communication flow such as via email, telephone
    - And for flow of artifacts (e.g., order form)

Example flow model for ticket buying

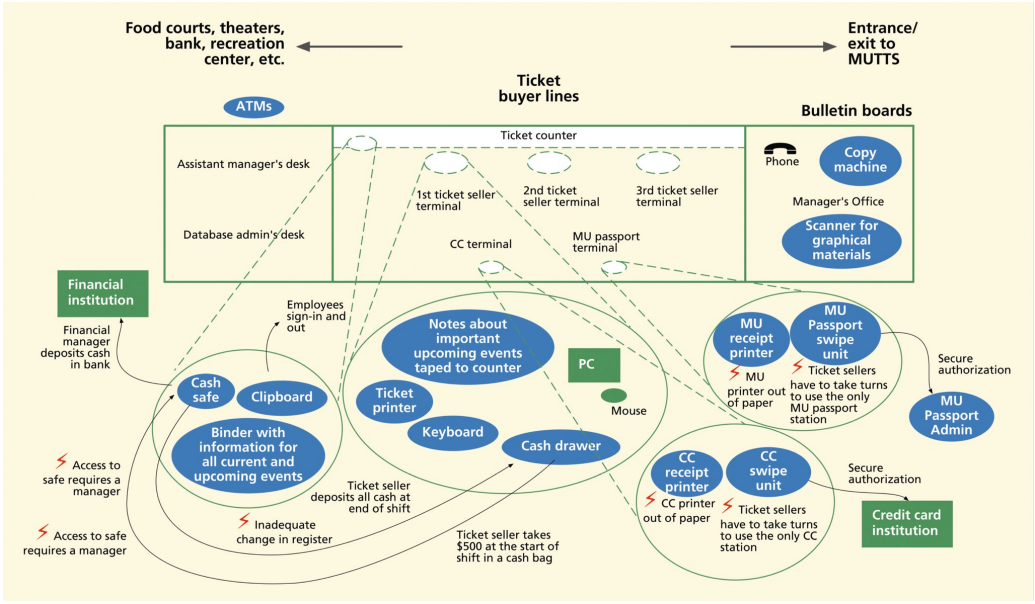


Example: Flow model with focus on artifacts (e.g., for restaurants)



Physical model

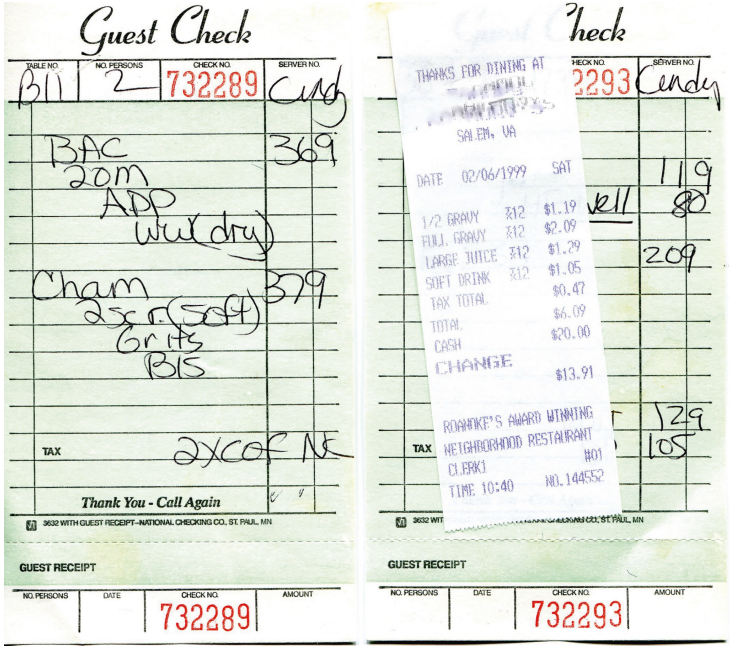
- Pictorial representation of physical layout of workspace
- Gives roles, activities, and artifacts of other models a physical setting
- Shows physical dimensions of workspaces, buildings, walls, rooms
- Shows personnel, workstations, physical equipment, communications, and collaboration spaces



Example: physical model of ticket buying

Artifact model

- One way to model data is to use artifacts
- A collection of artifacts used in the work practice (collected in data elicitation)
  - Work practice forms (e.g., order forms)
  - Props
  - Receipts
  - Memos
  - Correspondence forms
  - Templates for work products
- Show how **tangible elements** (physical or electronic) are used and structured in business process flow of doing work



- Example: Artifacts from a restaurant

Models for users and task structure

- If there is a broad range of user characteristics for any work role, consider making a user *persona*
- If there is a large number of user tasks, organize them with a task structure model, such as a hierarchical task inventory
- If some user tasks are a bit complex, describe them with *task sequence models*

User class definitions

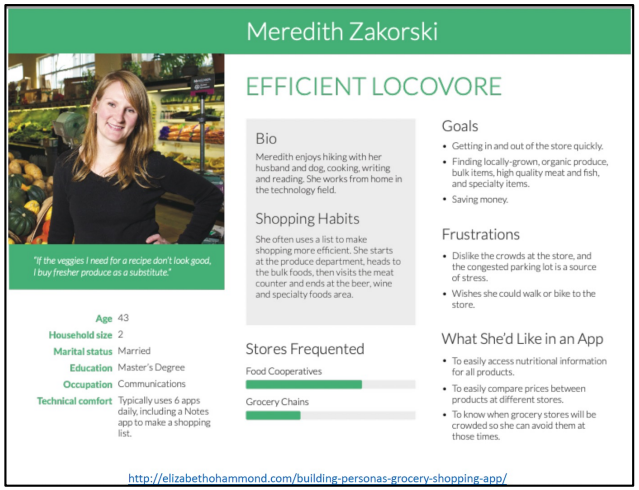
- Characteristics of a potential user community that can perform a given role
- Defined in terms of:
  - Demographics
  - Skills, knowledge, special needs
  - General public ticket buyer for Ticket Kiosk System could include:
    - Busy professionals getting tickets at lunch (need to get through quickly)
    - Senior citizens with limited motor skills and some visual impairment
    - Tourists from out of town to see a local show (some may speak different languages)

Personas

- Precise descriptions of a user and what they wish to do with a system
  - They are not real but are imaginary examples of the real users that they represent (e.g., from gathering information about the user types)
  - Talk about them more from just the perspective of user types – use concrete examples and real scenarios
  - Usually a .5 a page to 1 page long (otherwise too long)
  - Focus on details that allows someone to relate to them, understand “where they are coming from”, and identify potential issues

Personas

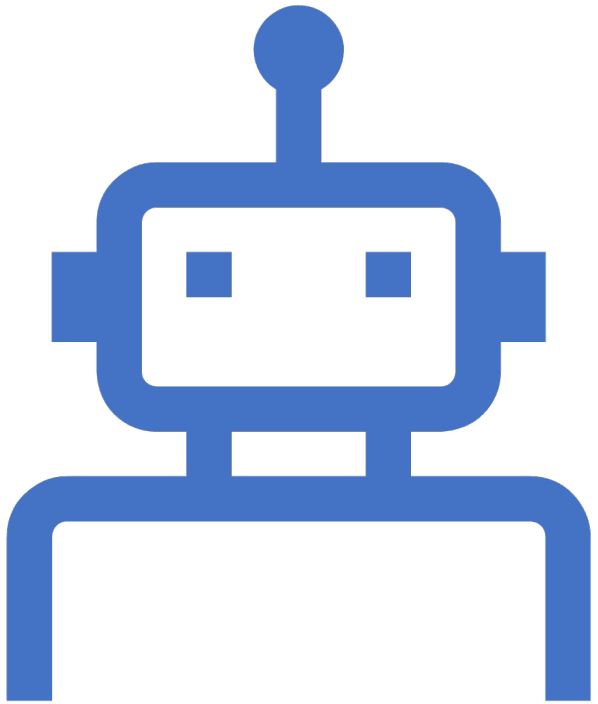
- In defining a persona:
  - Be as specific about the made-up details
  - Give the persona a name and an image
- Usually have a cast of about 3 personas with one or two being the prime personas



User personas (see slides from lab)

- Not an actual user
- Hypothetical, but specific, person to use as a particular design target
- A single user with very concrete characteristics, not an "average" user
  - Specific individual with a name, life, and personality

- A persona is a story built up from user research data
- Makes analysis and design discussions much easier
- Post user personas in your design studio
- Later: Using personas in design



Personas

- When you design the system, the design team will refer to the persona's name rather than just 'the user' – gives more personal connection
- It helps focus your design process on something realistic and limits the infinity of possibilities

Abby Grey

"I should make the most of my time while I'm still here on campus."

MAJOR

Mechanical Engineering

University

Dalhousie University

AGE

21

Bio

Abby is an ambitious and hard-working 4th year student. Born and raised in small town in NS, she moved to Halifax to pursue her dreams of engineering and has lived on campus throughout her university experience. She works part-time at a local coffee shop but makes sure to have enough time for her friends and social life.

Abby enjoys reading and hiking. She's active on social media. She likes using software in her courses but is a bit hestiant learning new technology in case she can't learn it quickly.

Wants & Goals

- Wants to graduate with excellent grades
- Wants to be more physically active and eat healthier
- Aspires to travel the world but feels unsafe travelling alone

Personality

Planner — Spontaneous

Thinking — Feeling

Extrovert — Introvert

Analytical — Creative

Frustrations

- Job searching
- Student loans
- Fear of underdeveloped social skills
- Useless mobile apps that take up space on her phone

- What are the important bits that you can pull from this persona that you as a designer would have to consider?

- Abby is a Engineering student at Dal
- She is active an likes reading
- Abby is an analytical thinker who likes to plan things out
- She is comfortable with technology but worries that it will take her time to learn new technology

- From this persona what are her goals:

- She wants to be healthier (both physically and eating)
- She wants to graduate with strong marks
- She wants to travel the world but is a bit scared to do so alone

- What are some factors that you might consider when designing the application using this persona?

- Make sure the apps are useful to her
- Give options to go back (undo) with helpful messages (no blame game) and provide a UX that will promote easier learning (eg., recall)
- Provide an emotional connection (feeling more than thinking)

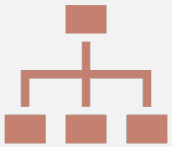
Task models

Task structure models

Task sequence models

Task structure model: Hierarchical task inventory

Decompose major tasks into multiple possible sub-tasks

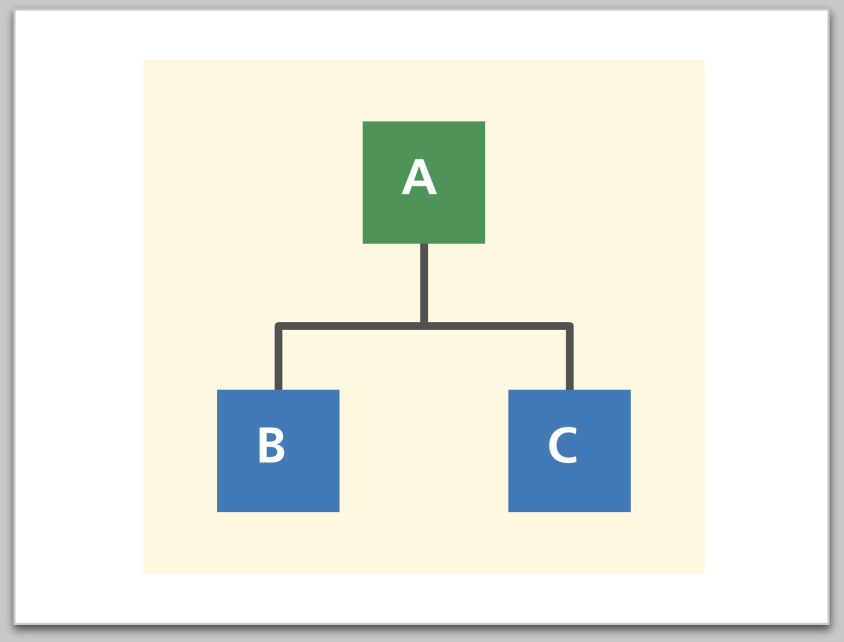


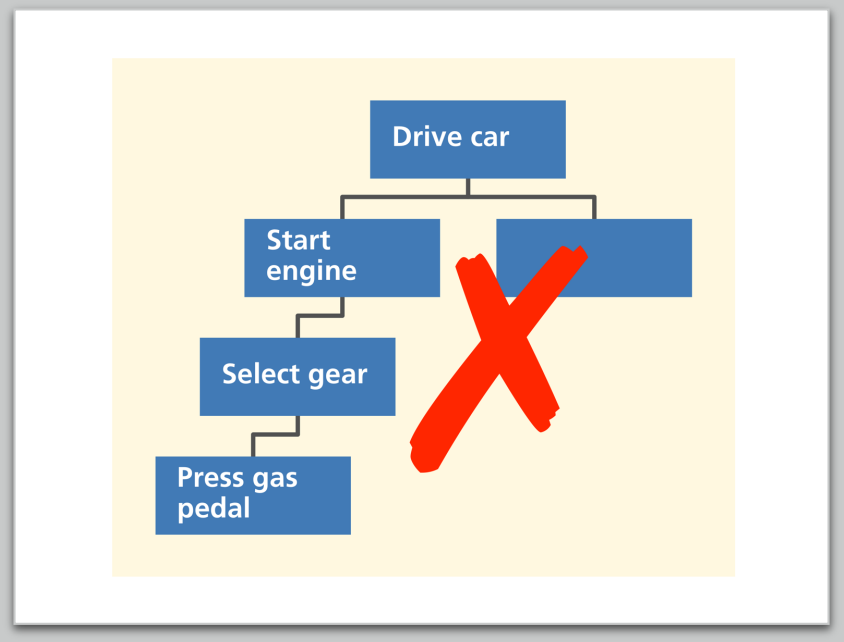
Just like functional decomposition in software hierarchy

Good way to visualize what user tasks and actions are possible

Hierarchical relationships

- If task A is immediately above Task B in the HTI
  - Task A is a super-task of task B
  - Task B is a sub-task of task A
- Avoid temporal relationships, sequencing with a hierarchical relationship diagram
- Instead think of tasks ands sub-tasks from the point of view of a user



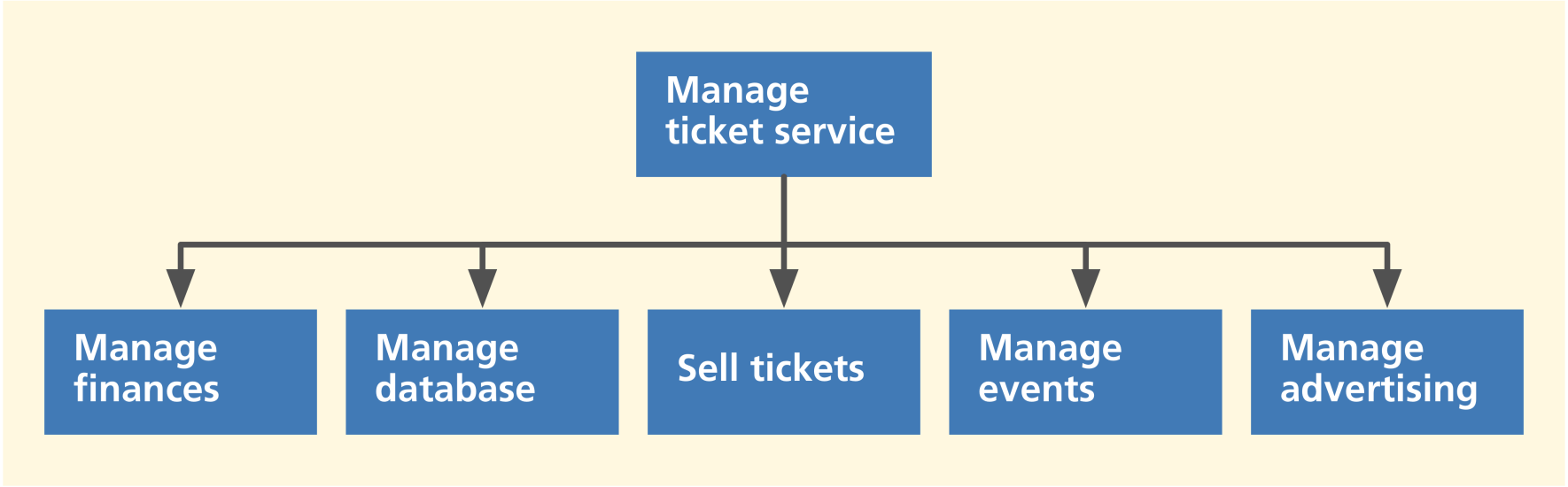


Hierarchical task inventory

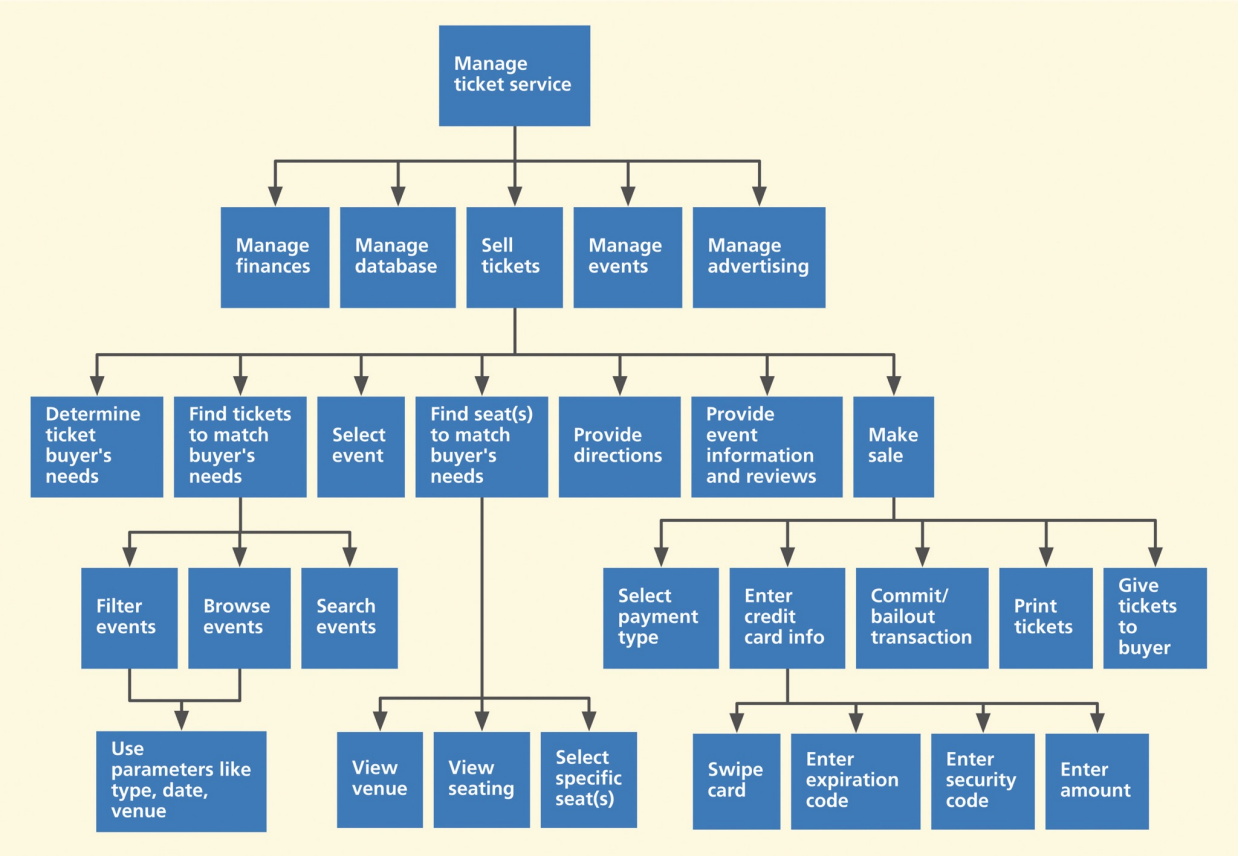
To guide overall design

Use as checklist for keeping track of task coverage in design

For matching that coverage to your inventory of scenarios and other task representations



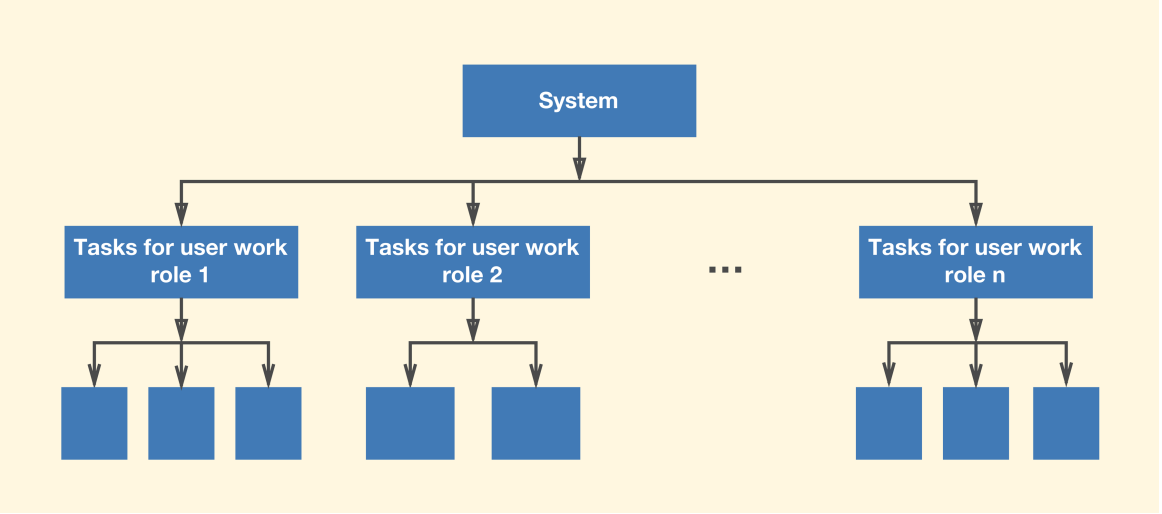
First-level HTI for ticket selling

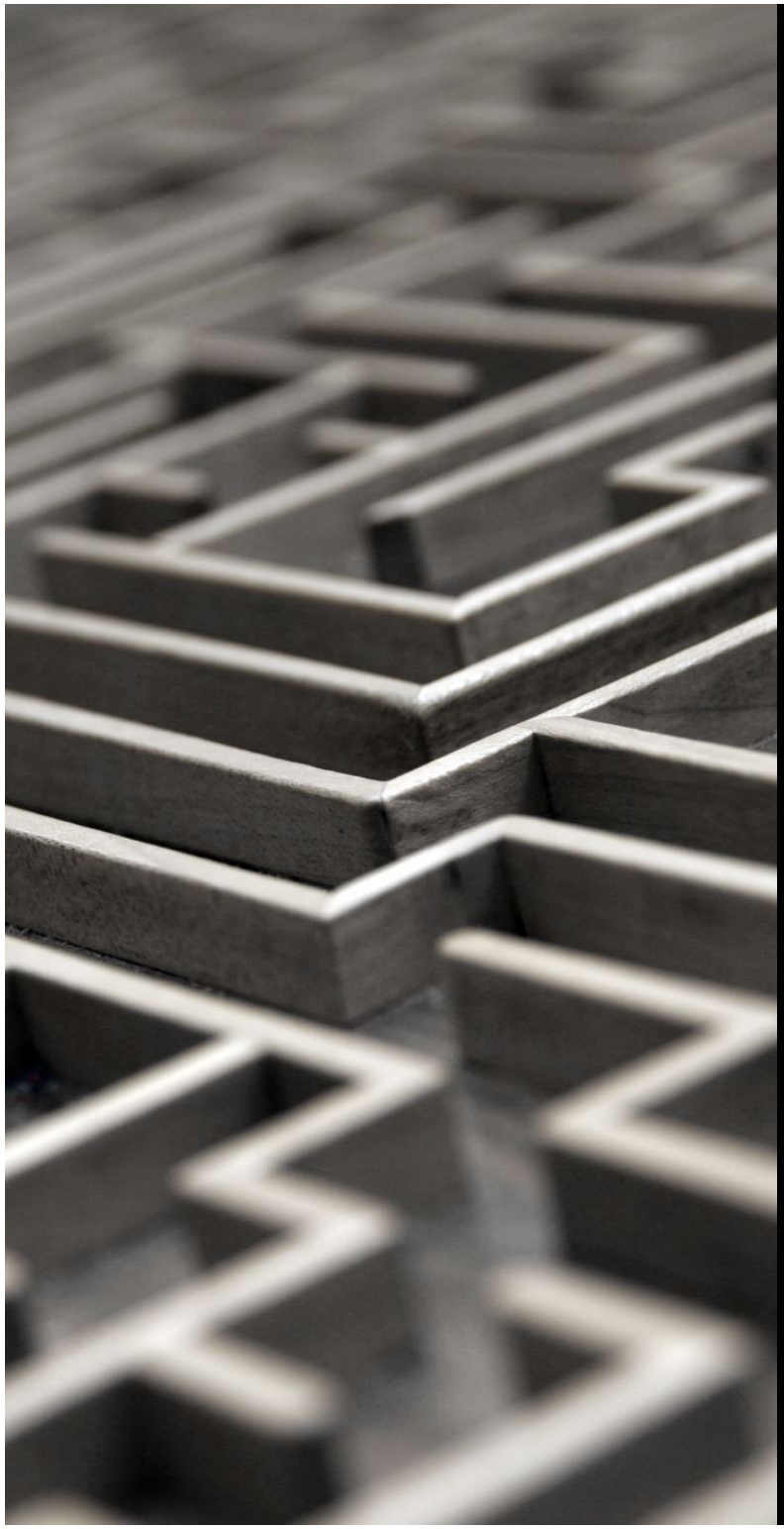


Example HTI for Ticket Selling

Try separating overall HTI by user work roles

- Full HTI can be too large to handle
- Tasks performed by one work role are usually separate from those of other work roles
- Can do analysis and design for each user work role separately





Task sequence models

- Task sequence: Step-by-step description
- Often has user actions and system responses

Task Analysis Techniques

- Describes what is to be done
  - Focus on the steps to complete a task
- Describes how a user does the task
  - This captures the knowledge people need to do the task
  - Also, focuses on what they actually do to the do the task

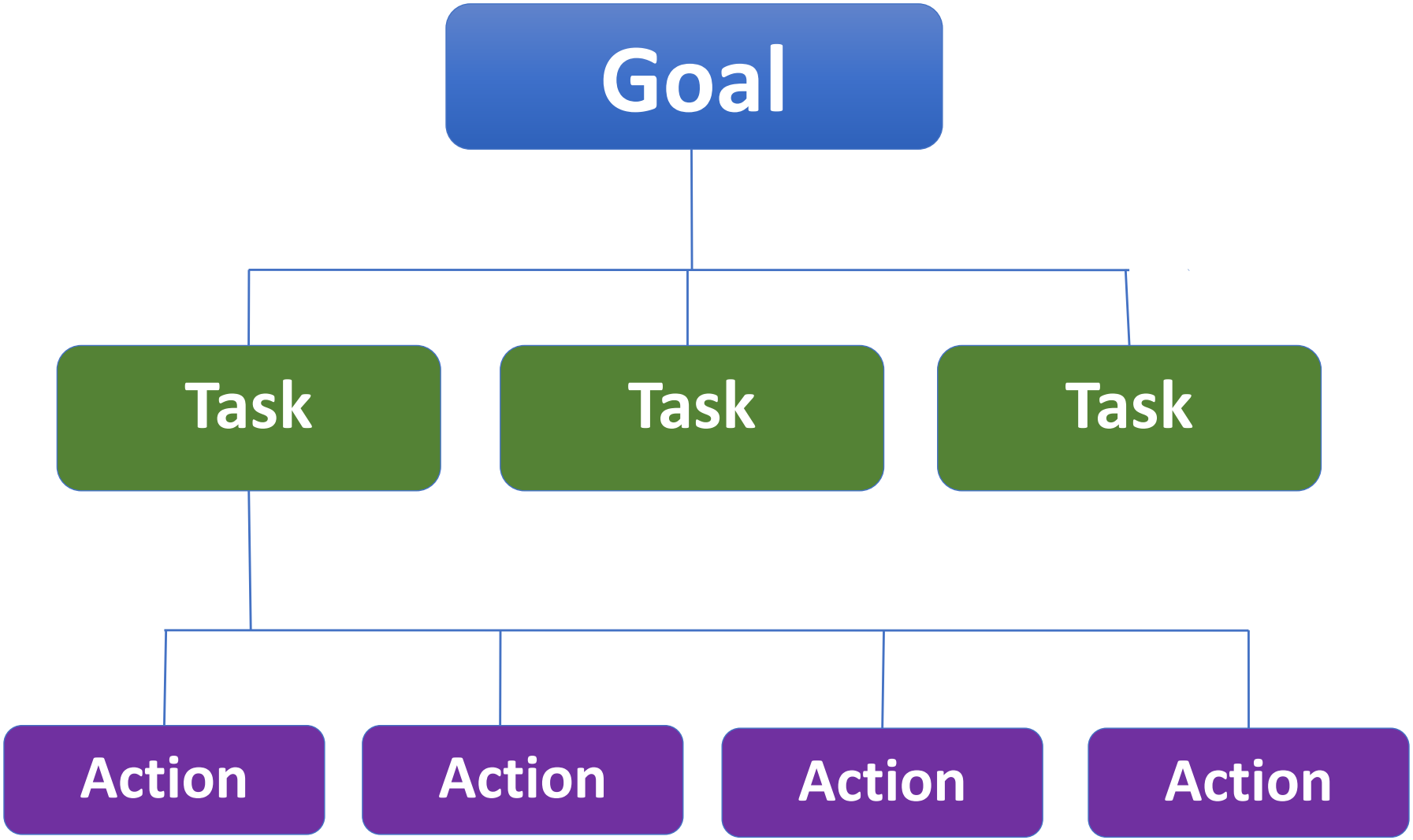
Should do vs. actually do

- There can sometimes be a difference in what the steps *should be done* to do a task and what the user *actually does* to perform the task
  - This difference can suggest inefficiencies with a system or indicate need for improvements to the UI or new system/support to be added
  - Sometimes users adapt the the steps or do “walk arounds” (e.g., use Google to refind pages) to perform tasks or use outside “artifacts” to help (e.g., sticky notes)

Goals, Tasks & Actions

Often when we talk about design, we refer to user goals, their tasks, and the actions

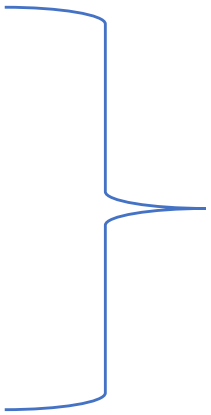
- Goal:
  - The end result to be achieved
  - High level that states *what* is to be achieved not *how* it is to be achieved
  - Can be accomplished by performing a set of tasks
- Task:
  - A structure of set of related activities undertaken in some sequence
  - It what needs to happen to achieve a goal
  - At some point, the user will need to interact with a device to do some actions related to the tasks
- Action:
  - Individual operations or steps that need to be undertaken to complete the task


Goal
Task
Task
Task
Action
Action
Action
Action

Example

- Goal: Sally wants to communicate to her friend Gail in writing

- This can be done in several ways:
  - Handwritten note/letter
  - Email
  - Text message



Each require different set of tasks and actions

- Handwritten note/letter:
  - Tasks:
    - Obtain paper and a pen or pencil
    - Find a flat surface to write on
    - Use pen/pencil to write words
  - Actions:
    - Write individual letters of alphabet to makeup words

Goals, Tasks and Actions

- Goals and tasks and actions may be different for different people depending on their previous knowledge and experience
  - Eg. Actions in particular may be different between novice and experienced users
    - E.g., using control short-cut keys
- By focusing on user goals and their tasks and actions you can design systems that reflect what users want to do…

Task sequence model components

- Task and step goals
- Steps
- User actions
- Task triggers
  - What happens to cause users to initiate each task or step?
  - Example, incoming phone call leads to filling out order form

Scenario: Simplest task sequence model

- Narrative description of task performance
  - Stories about
    - Specific people performing work activities
    - Specific existing work situation/context
    - Told in a concrete narrative style, as if it were a transcript of a real usage occurrence
  - Deliberately informal, open-ended, and fragmentary

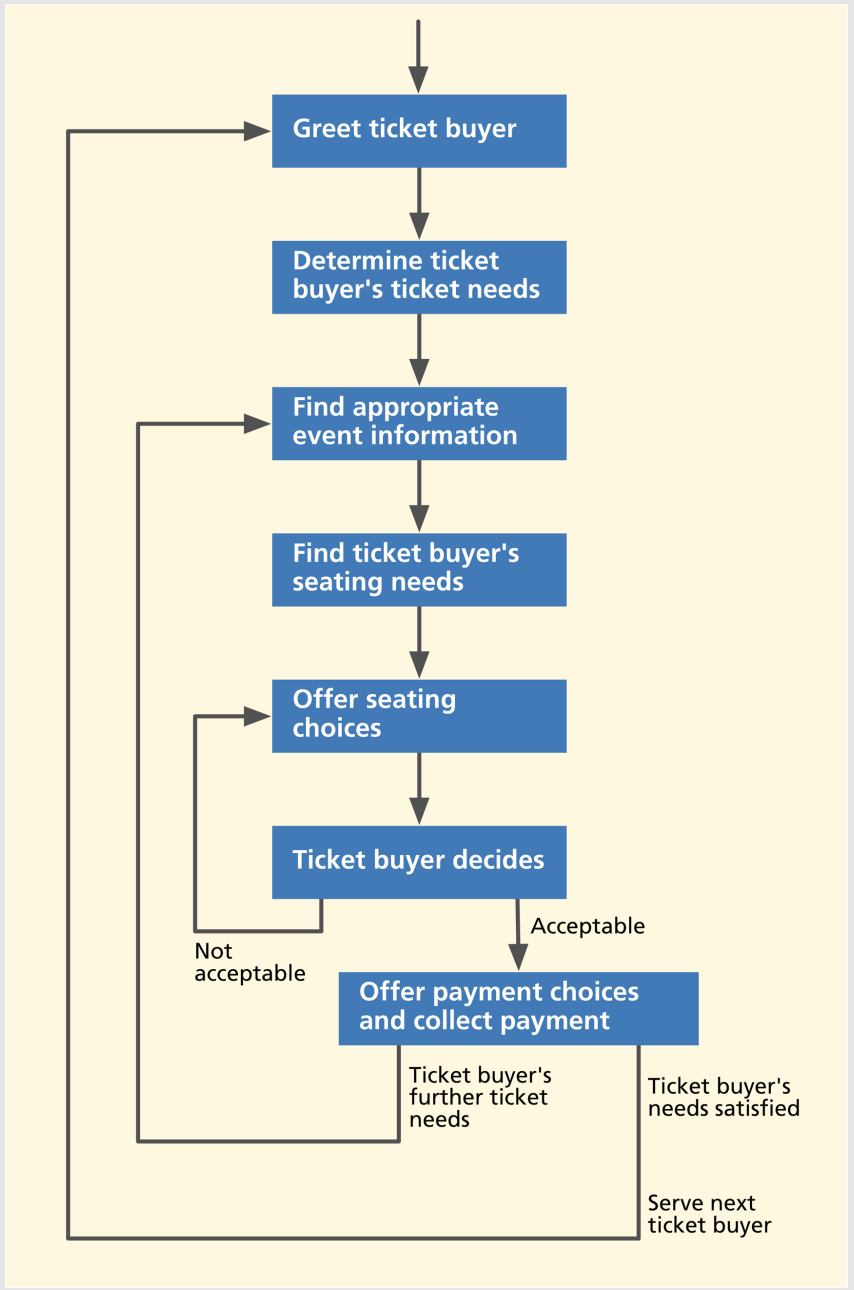
Step-by-step task sequence models

- Contains a detailed description of task performance observed in users or as told by users
- Shows detailed steps of task performance
- Includes temporal ordering of actions and activities

ticket buyer	ticket seller
1. Wait in line until turn	2. Greet ticket buyer
3. Describe event for needed tickets	4. Look up event in database. How many tickets will be needed?
5. State number of tickets	6. Look up venue seating chart. Describe available options by seat location and price
7. Select seats	8. Calculate and state total cost. How would you like to pay?
9. Pay with credit card	10. Give tickets, receipt, and card to ticket buyer

Simple ticket transaction

Branching and looping in ticket selling example



Essential **Use Case** Task Sequence Models

- Credit due to researchers: Constantine and Lockwood
- A structured narrative
- In language of users in work domain
- Describes task with a single user intention or goal
- Complete, well-defined task description that is meaningful to a user in a role
- Abstracted to the essentials
- Does not mention specific technology or user actions (e.g., clicking on a button)

Example, Essential Use Case

- Paying for a ticket purchase transaction (with a credit or debit card)
- Sequence of user intentions and system responses

User tasks vs. system functions

- Task name should be something a user will do, not a system function
- In contrast, a function is something system does
  - Example: Information is viewed/displayed

Example, essential use cases

Notice there is a 'user' and a 'system' in the case study

1. **Ticket buyer:** Express intention to pay
2. **System**: Request to insert card
3. **Ticket buyer:** Insert card
4. **System**: Request to remove card quickly
5. **Ticket buyer**: Withdraw card
6. **System**: Read card information
7. **System**: Display summary of transaction and cost
8. **System**: Request signature (on touch pad)
9. **Ticket buyer**: Write signature
10. **System**: Conclude transaction
11. **System**: Issue receipt
12. **Ticket buyer**: Take receipt

Using Scenarios and Use Cases

- Scenarios and use cases are two methods used to describe tasks and the users performing them
- They encourage you to consider:
  - A wide range of users who will use the system
  - The range of activities that they system will be used for
  - And the environment that the tasks will be performed
- The process of writing these, will help you better understand your users and tasks
- They can be refined over time and used in later evaluations and usability testing during the design process

Scenarios

- Informal narrative description
  - Describes human activities or tasks in a story that allows for exploration and discussion of contexts, needs, and requirements
  - It does not describe the use of the software or the technical support to achieve the task
- Helps to understand why people do the things they do and what they are trying to do

Example of a Scenario reflecting a local fisher (used in a recent project)

Fisher Scenario

Bob is lobster fisher from the Eastern Shore of Nova Scotia. Bob comes from a long line of fishers and he himself has been fishing for about 30 years. During lobster season, Bob often depends on his own experience and knowledge that has been passed down to him by his Dad and Grandfather. Over the last few years, Bob has started to integrate technology such as a GPS, depth indicator, and fish sonar on his boat. While comfortable with technology, he needs to see value when adapting it for his livelihood. For example, on the Eastern Shore, it can get quite foggy and he appreciates having the GPS to help him navigate when the visibility is poor. It also helps him keep track of where he has set his 30+ traps when he checks on them.

Over his career, Bob has made a good living fishing from hard work and by being a keen business person. His main focus is on lobster when in season but does fish other varieties in the off-season. As such, Bob always wants to make sure that his 47 foot boat is in good condition. Also, as Bob ages he is more conscience of his own health and by keeping the boat in good working order it makes it easier to fish and to deal with the physical requirements of the job.

The last couple of years his daughter Sara has been helping out on the boat learning the trade. She recently graduated university with an environmental degree and want to find ways to make it easier to fish, to make the fishing industry greener, and to maintain the sustainability of the fish stock. She is always researching new technology (e.g., lobster trap tracking devices) to see how it could be used on their boat. Bob and she don't always agree on the technology. But they both agree that it makes sense to decrease the fuel cost. This is a large running expense of the boat which Bob would like to reduce, and Sara sees the environmental benefits of this.

Scenario Example: A Travel Organizer

"The Thomson family enjoy outdoor activities and want to try their hand at sailing this year. There are four family members: Sky (10 years old), Eamonn (15 years old), Claire (35), and Will (40). One evening after dinner they decide to start exploring the possibilities.

They all gather around the travel organizer and enter their initial set of requirements – a sailing trip for four novices in the Mediterranean. The console is designed so that all members of the family can interact easily and comfortably with it. The system's initial suggestion is a flotilla, where several crews (with various levels of experience) sail together on separate boats. Sky and Eamonn aren't very happy at the idea of going on vacation with a group of other people, even though the Thomsons would have their own boat.

The travel organizer shows them descriptions of flotillas from other children their ages and they are all very positive, so eventually, everyone agrees to explore flotilla opportunities. Will confirms this recommendation and asks for detailed options. As it's getting late, he asks for the details to be printed so everyone can consider them tomorrow. The travel organizer prints out a summary of the different options available."

Scenarios

- Telling a story is a natural way for people to describe how to do something
- Helps the design team relate to the user tasks and helps the team to grasp and convey design ideas before starting on the design
- Inclusion of emotional elements (not over the top) can help improve the designer's appreciation of the user experience
  - And when used with a combination of personas it can improve the designer appreciation of the user experience even more
  - Persona: provides the attributes of a person, aspects of their personalities and needs
  - Scenario: describes the activities and context of tasks, can even include a “day in the life of a person”

Scenarios

- Can be used to imagine potential uses of a system and to capture existing behaviour
- Can be used to describe existing behaviour before a system and then futuristic system and how it could support the user in the same tasks

Use Cases

- Like scenarios, use cases also focus on goals but more on the user-system interaction rather than just the user tasks
- There is a **user** and it is how the **system** can help the actor achieve their goal
- It is used to envision what the future system will be like to the user
- There are two parts:
  - Normal course
  - Alternative course

Use Cases

- Normal Course
  - This is the main use case
  - It is a set of actions that the analyst believes to be the most commonly performed by the actor
  - It is the normal case without exceptions (ie nothing goes wrong)
  - Use cases number the steps from 1, 2, 3, … n
- Alternative Course
  - These include any possible sequences that could happen outside of the normal course (e.g., exceptions, errors, etc)
  - These are listed after the Normal Course of actions
  - In the Alternative Courses, you replace the number n with a sequence of alternative steps n, n.1, n.2, … n.m
    - For example if the alternate course happens at step 2 of the normal course then the numbering for the alternative course would be 2.1, 2.2, etc.

Use Case Example for the Travel Organizer

TASK: Finding out visa requirements for travelling

Normal Course

1. The system displays options for investigating visa and vaccination requirements.
2. The user chooses the option to find out about visa requirements.
3. The system prompts user for the name of the destination country.
4. The user enters the country's name.
5. The system checks that the country is valid.
6. The system prompts the user for her nationality.
7. The user enters her nationality.
8. The system checks the visa requirements of the entered country for a passport holder of her nationality.
9. The system displays the visa requirements.
10. The system displays the option to print out the visa requirements.

Normal Course

1. The system displays options for investigating visa and vaccination requirements.

2. The user chooses the option to find out about visa requirements.

3. The system prompts user for the name of the destination country.

4. The user enters the destination country's name.

5. The system checks that the country is valid.

6. The system prompts the user for her nationality.

7. The user enters her nationality.

8. The system checks the visa requirements of the entered country for a passport holder of her nationality.

9. The system displays the visa requirements.

10. The system displays the option to print out the visa requirements.

Alternative Courses

6. If the destination country name is invalid:

6.1 The system displays an error message.
6.2 The system returns to step 3.

8. If the nationality is invalid:

8.1 The system displays an error message.
8.2 The system returns to step 6.

9. If no information about visa requirements is found:

9.1 The system displays a suitable message.
9.2 The system returns to step 1.

Moving across the gap from analysis to design

- From existing work/problem domain and work practice
- To envisioned design domain and work practice
- Usage research data guide and inform design but the design isn’t explicit in the data
- This is “the step where magic happens”

Design thinking

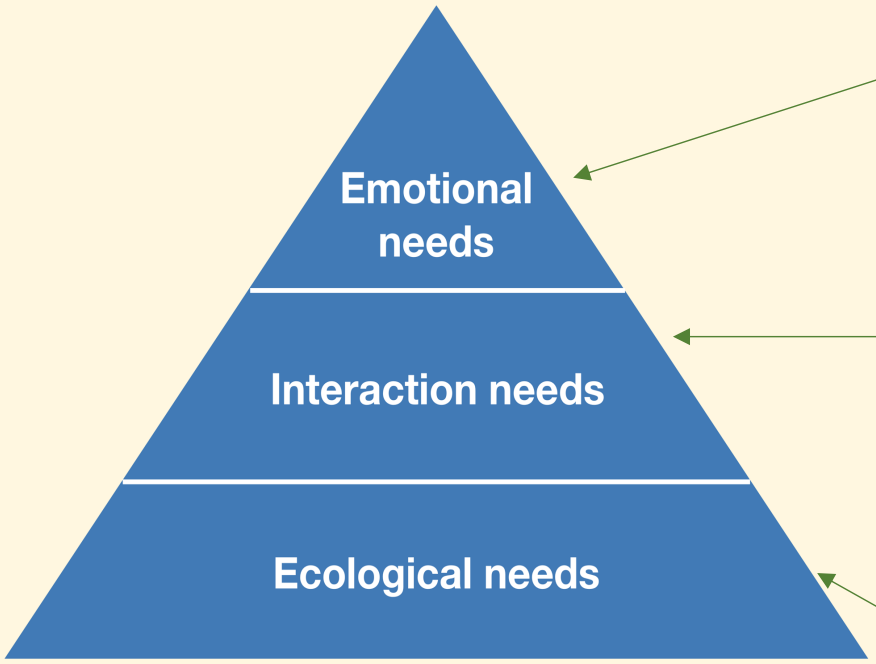
- Design thinking is seeing everything in terms of design
- Design thinking is immersive
- Design thinking is integrative
  - Brings together usage research, analysis, modeling, creativity, innovation
- Design thinking is market-oriented (think Apple products)
  - When I got my iPad, label didn't say “Made by Apple”
  - It said, “Designed by Apple!”

Design as a noun and as a verb

- Design as a noun: The **concept or plans** for a product or system
  - A designer's conception and description
- Design as a verb: The **act** of creating a new design solution
  - The Design Solutions box in the UX lifecycle process

Design to satisfy human needs

- Pyramid of human needs in UX design
  - In the context of a work practice
  - Layers are filters to guide thinking, scoping, discussing, and doing design



Users need to be emotionally and culturally satisfied and enriched as they use the product

- About aesthetics and joy of use
- About emotional impact and value-sensitive aspects
- About social and cultural implications

•About how users operate system or product
•A task, intention, and user action view
•Where users look at displays and manipulate controls (e.g., sensory, cognitive, and physical actions)

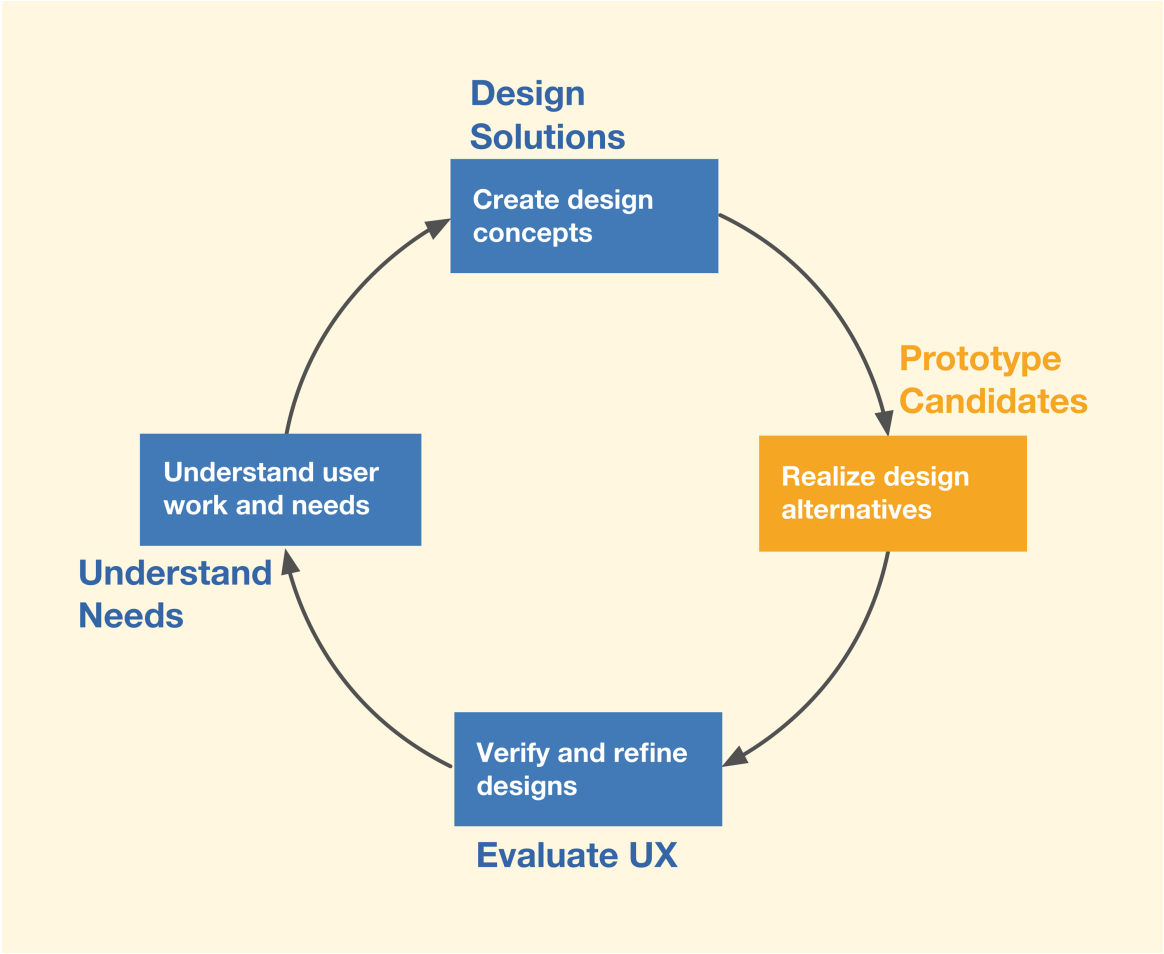
•About how system or product interacts or communicates with people and other systems in its environment/context
•Example: An iPod and the complicated iTunes ecology

Ideation iteration

- Lightning-fast
- Loosely structured iteration
- For purpose of exploring design ideas
- Role of prototype played by sketches (and others like storyboarding)

Prototyping

Prototyping: Affords a way to evaluate design before it's too late to change it without great expense



Prototypes

- Realize design alternatives
- To envision promising design candidates
- Tangible design representations that can be evaluated with users
- Create representations of design to required fidelity
- Like sketching, often done in parallel with design

Prototypes

- Concrete baseline between designers and users
- Conversational props
- Something for user to take for a spin
- Visibility to stakeholders
- Impression that it’s easy to change (obviously unfinished)

Conceptual design: from requirements to design

- Transform user requirements/needs into a conceptual model
- "a description of the proposed system in terms of a set of integrated ideas and concepts about what it should do, behave and look like, that will be understandable by the users in the manner intended"
- Don’t move to a solution too quickly. Iterate, iterate, iterate
- Consider alternatives: prototyping helps

Definition of Prototype

- What is Prototype?

  A Prototype is a limited representation of a design that allows users to interact with it and to explore its suitability

  For example, it could be:

  - a paper based set of screens (e.g., GUI or web site)
  - 3-dimensional paper/cardboard mockup of a work station
  - Stack of hyperlinked screen shots

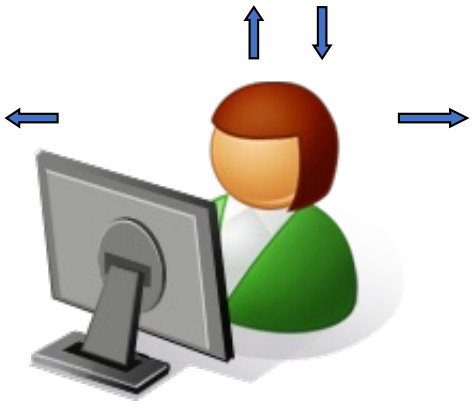
Stone, D., Jarrett, C., Woodroofe, M., Minocha, S. (2005). User Interface Design and Evaluation: Morgan Kaufmann, California.

Sharp, Rogers, Preece. (2007). Interaction Design: beyond human-computer interaction (2nd edition): John Wiley & Sons, England.

For example, Jeff Hawkins when developing the palm pilot carved a piece of wood to the size that he envisioned and carried it around and pretended to add information to it…

He did his to discover ways in which the product might be used (that he and his team had not thought of and to see whether the physical prototype was a suitable for the envisioned tasks)

•Similar to the researcher who wanted to know if a mouse controlled by nodding your head would be a good idea… do you think it was??



Handspring Visor

Purpose of prototyping

- To test technical feasibility of an idea
- To communicate a device among your team members
- To use when discussing the product's need with the client
- To clarify vague requirements
- To test different ideas with users (user evaluations)
- You can test out ideas for yourself
- It encourages reflection: very important aspect of design
- To answer questions and support designers in choosing among alternatives

Purpose of prototyping

- The purpose for prototyping will influence the type or prototype you build.
- For example, if you were trying to clarify how users might perform a set of tasks and whether your design would support this, you might produce a paper-based mockup

How do we do this??

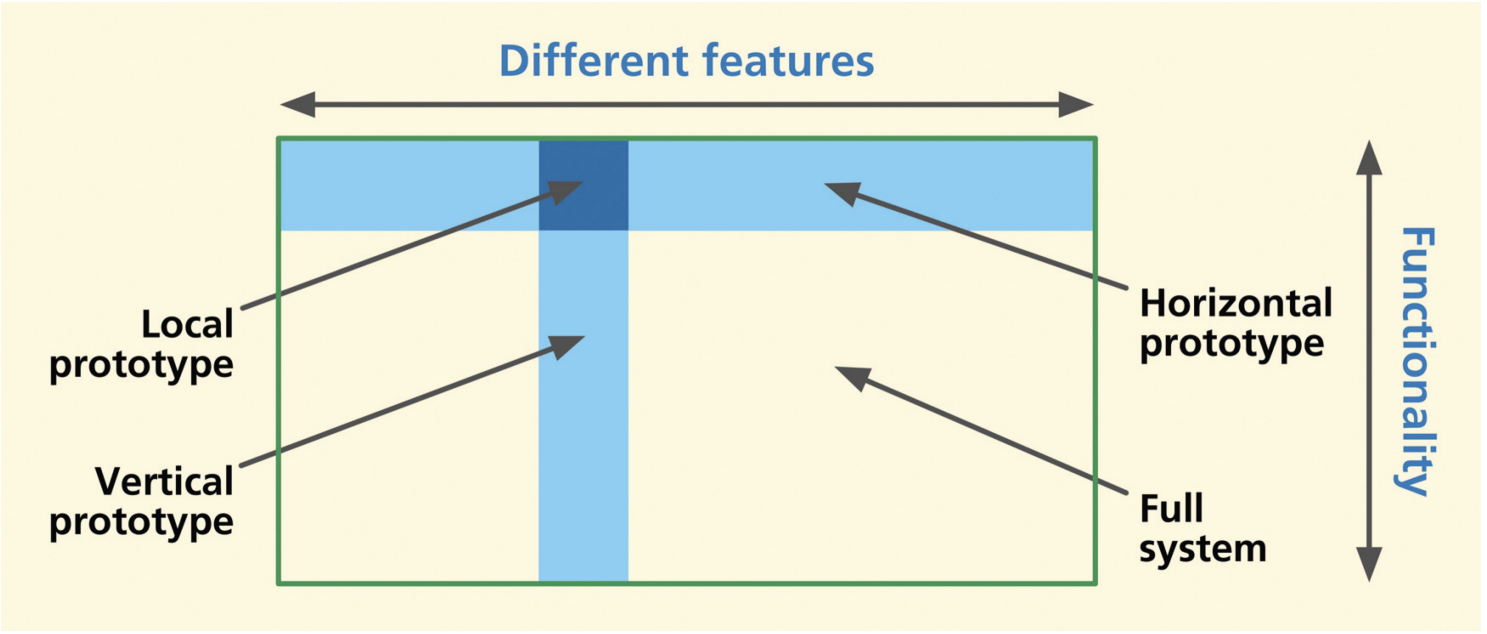
- Don’t move to a solution too quickly. Iterate, iterate, iterate
- Consider alternatives: prototyping helps

Choosing right kind of prototype

- Choosing the right
  - Breadth
  - Depth
  - Level of fidelity
  - Amount of interactivity
- Textbook has a good description of this

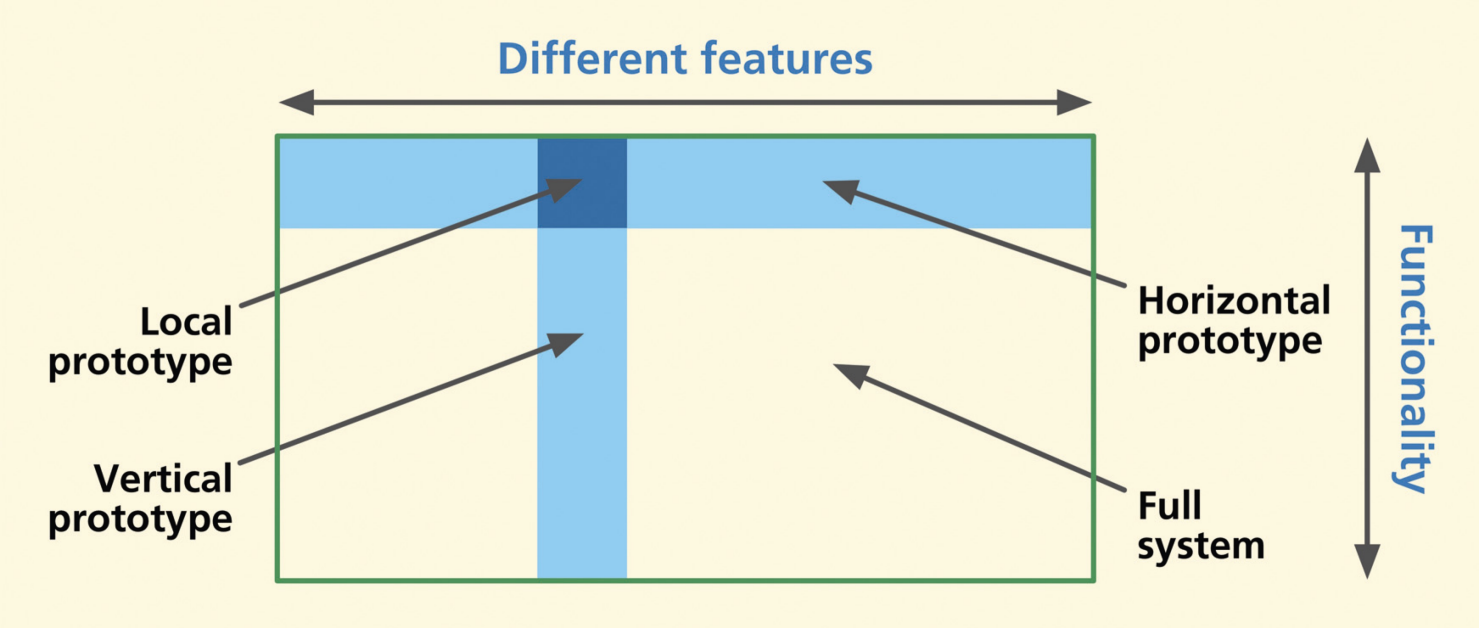
Horizontal prototype

- Breadth-wise slices of features
- More features
- Less detail per feature
  - Offers overview
  - Top-down
  - To demo product concept



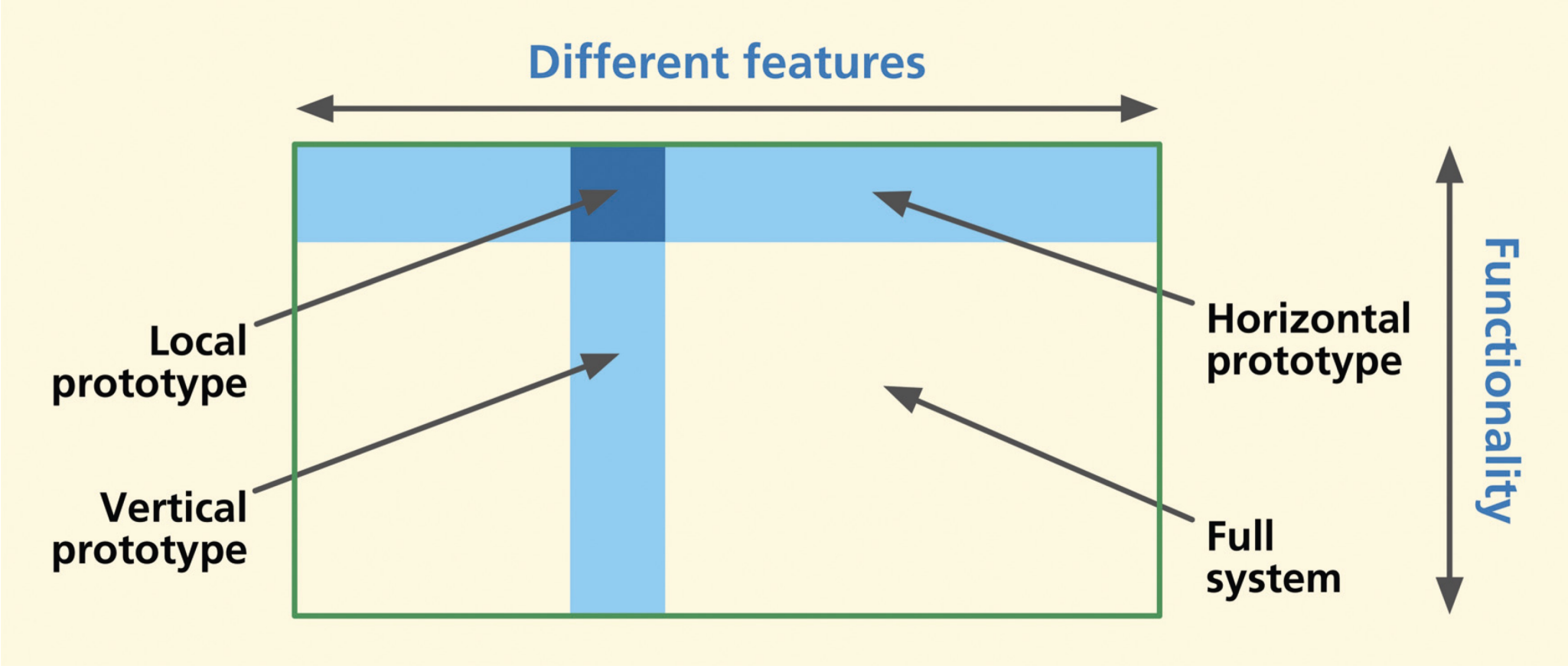
Vertical prototype

- Depth-wise slices of features
- Fewer features
- More detail per feature
  - Can even connect to some real functionality
  - UX evaluation can be more realistic (for the features covered)
  - Used to study a particular tasks sequence



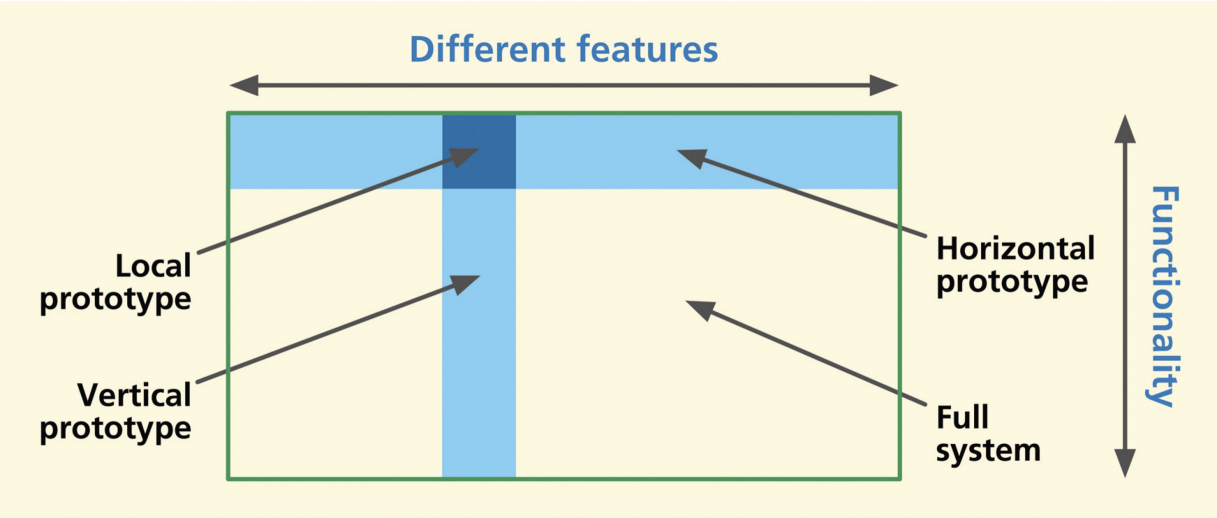
"T" prototype

- Most of user interface realized at shallow level (horizontal bar of “T”)
- A few parts done in depth (vertical stems of “T”)
- Nice balance, advantages of both



Local prototype

- Narrow in both dimensions
- Small area where horizontal and vertical slices intersect
- Used to evaluate design alternatives, focusing on a localized interaction design issue
- For particular isolated interaction details (e.g., of a dialogue box)
- Examples
  - Appearance of an icon
  - Wording of message
  - Behavior of individual function
  - Good for when your design team just can't decide a small part of design



Fidelity of prototypes

- Fidelity of content and appearance
- Reflects how “finished” prototype is perceived to be by customers and users
  - Not how authentic or correct underlying code is

Low-fidelity prototypes

- Faster, less cost, more flexible
- Low fidelity can mean paper prototype, but in agile UX it means simple wireframe prototype
- Not faithful representations of details of look, feel, and behavior
- Give rather high-level, more abstract impressions of intended design
- For exploration to find the right design

Low-fidelity prototypes

- Appropriate for early project stages when design details
  - Have not been decided
  - Are changing the most
- Proven that test users can take them seriously
- Proven effective in design evaluations

Low Fidelity Prototyping

- Does not look very much like the final product
- It uses materials that are different from the intended final version (e.g., the wooden palm pilot)

Low Fidelity Prototyping

- Are useful because:
  - Simple
  - Cheap
  - Quick to produce
  - Rapid feedback
  - Quick to modify
  - Allows designers to explore alternative designs and ideas (important in early stage of development)***
- They are not meant to be kept or integrated into the final product (only for exploration)

http://www.sapdesignguild.org/community/book_people/review_paper_proto.asp

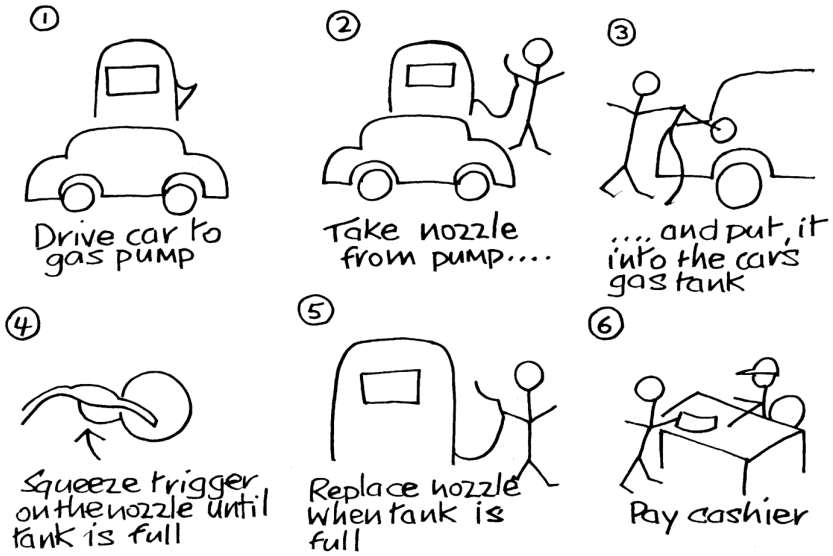
Types of Low Fidelity Prototypes

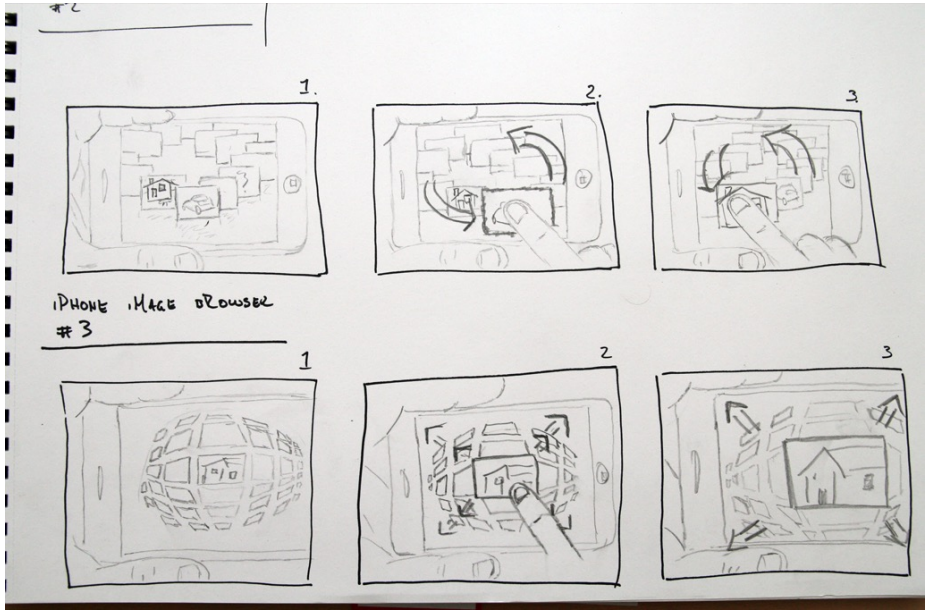
1. Sketching
2. Using Index Cards
3. Wireframes
4. 3-D Physical Mock-ups

5. Wizard of Oz

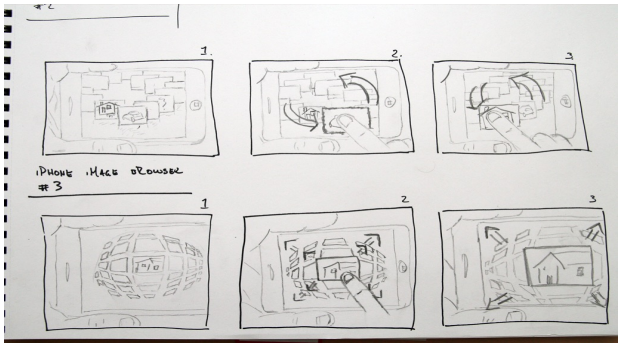
Sketching

- Relies on sketching (you don't need to be an artist!)
- Can use simple boxes, stick figures, etc.
- If you are drawing an interface, you could use different icons, dialog boxes, etc.





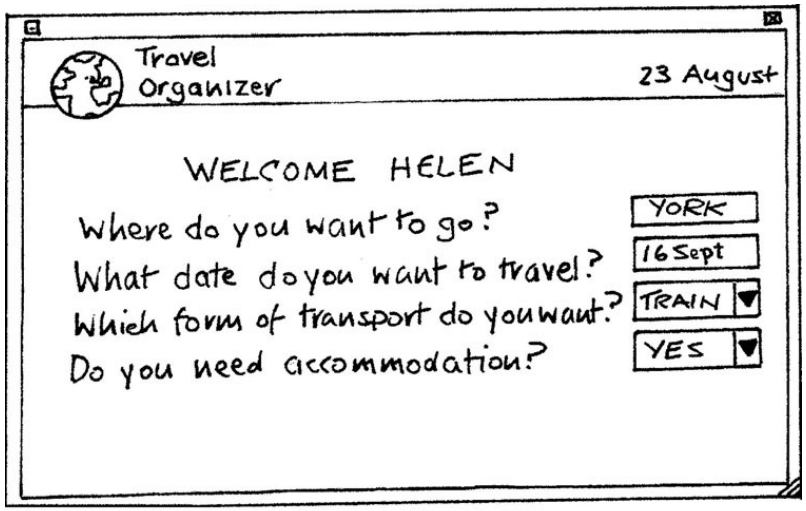
Adaptation to sketching: Animated prototypes

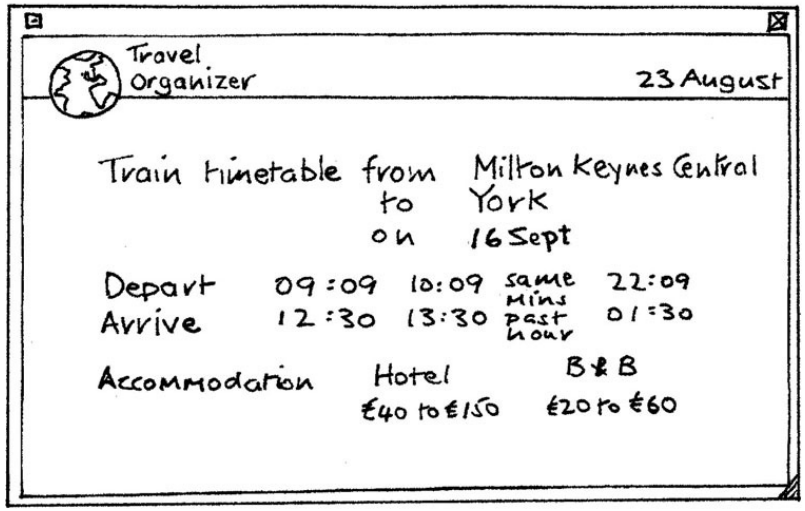


- Video animations, usually based on series of sketches
- Storyboard frames in “flip book” style sequence on video
- Can be very engaging and stimulating of discussion

Prototyping with Index Cards

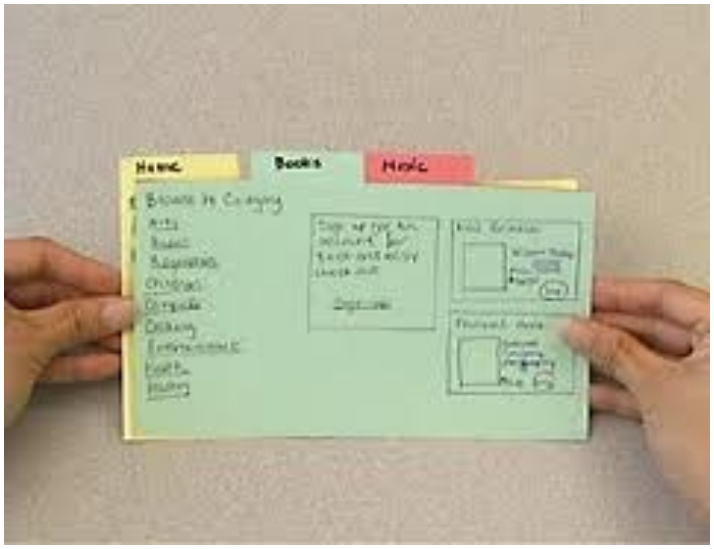
- Uses index cards (small pieces of cardboard about 3x5 inches)
- A successful and simple way to prototype an interaction
- Used quite often when prototyping a website
- Each card represents one screen or one element of a task
- User can step through the cards pretending to perform the task while interacting with the cards





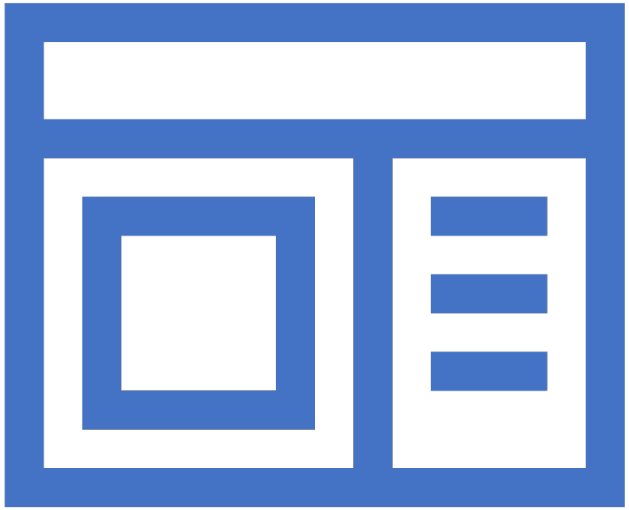
Advantages of using a card-based prototype

- The screens or screen elements can be manipulated and moved to simulate the interface (either with a user or without)
- Can be used to gain feedback and can be shown to colleagues to get another designers perspective



Google:
–https://www.youtube.com/watch?v=JMjozqJS44M
–Other examples
https://www.youtube.com/watch?v=x48qOA2Z_xQ
https://www.youtube.com/watch?v=yafaGNFu8Eg

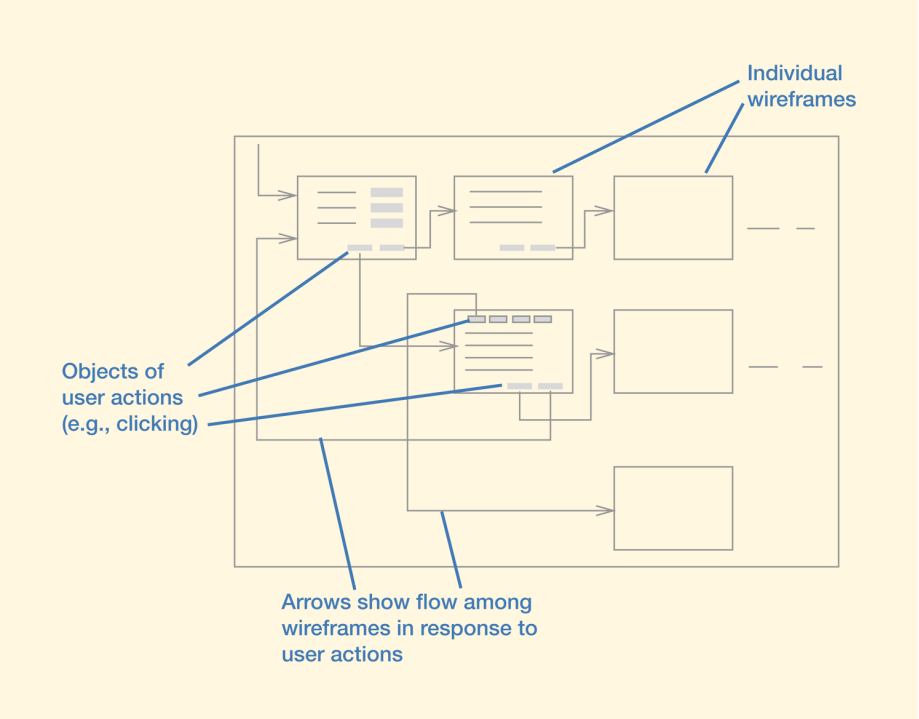
Wireframe prototypes



- The go-to prototyping technique of agile UX design
- Start as sketches in generative design
- As we have said, a wireframe is a two-dimensional drawing with lines, arcs, vertices
- “Widgets” like buttons, icons, menus, tabs, search box
- Textual labels
- Representing screen layout for interaction design
- Best done with a software tool (e.g., Sketch)
- Can also be done on paper

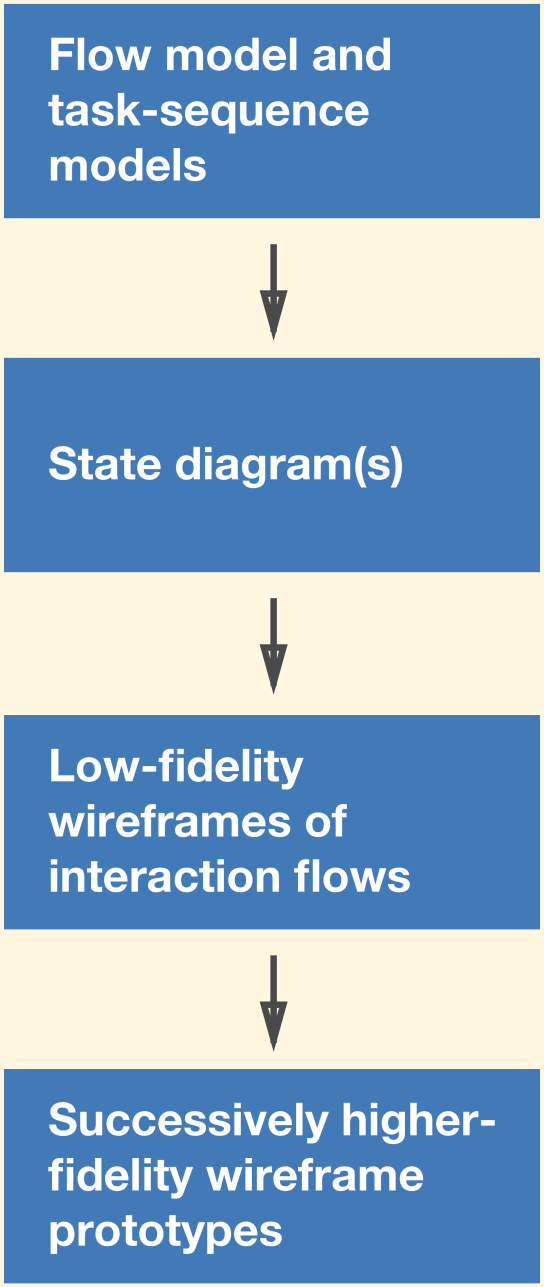
Wire flow prototypes

- Graphical representation of interaction/navigation flow
  - Wireframes as nodes of graph
    - Interaction states
  - Navigational transitions as arcs of graph
    - User actions (e.g., button click) as triggers



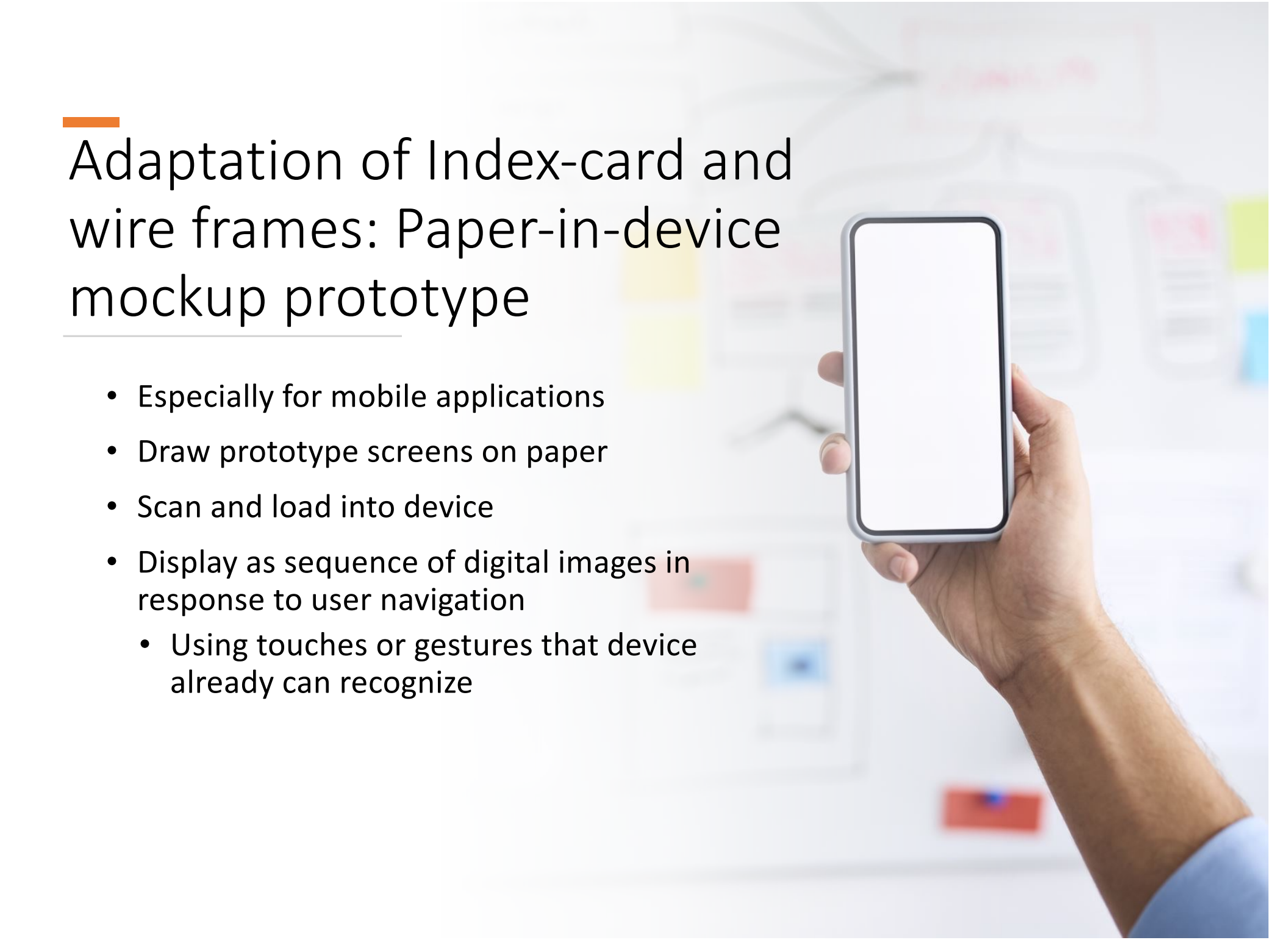
Wireflow prototype as a graph

- Focus on workflow
  - Start with task sequence models
  - State diagrams are graph form of workflow



Adaptation of Index-card and wire frames: Paper-in-device mockup prototype

- Especially for mobile applications
- Draw prototype screens on paper
- Scan and load into device
- Display as sequence of digital images in response to user navigation
  - Using touches or gestures that device already can recognize



Physical mockups

- A three-dimensional sketch
- Like sketches
  - Made quickly
  - Highly disposable
- To create with a tangible feel for exploring design visions and alternatives
- If primary characteristic of product or system is physicality

Physical mockups for tangible feel and for physical interactivity

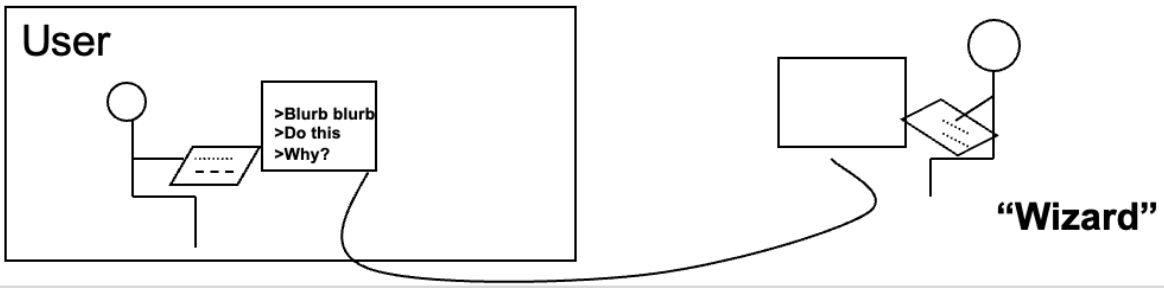
- Use materials at hand: cardboard, wood, or (e.g., a physical ticket kiosk mockup of cardboard boxes with an embedded touchscreen)
- Glue on simulated buttons
- Use real hardware controls such as push buttons, tilt buttons, sliders
- Example, knobs and dials, rocker switch
- Example, joystick from an old Nintendo game

Rough mockup of a "rickshaw"-style cart for transportation

Example, a more finished physical mockup

Wizard of Oz

- Assumes that you have a software-based prototype
- The user sits at the computer system (or uses one) and interacts with the computer screen and software as though interacting with the product
- Really the computer is connected to another machine where a human is operating it and simulating the software response to the user (or the software simulates)



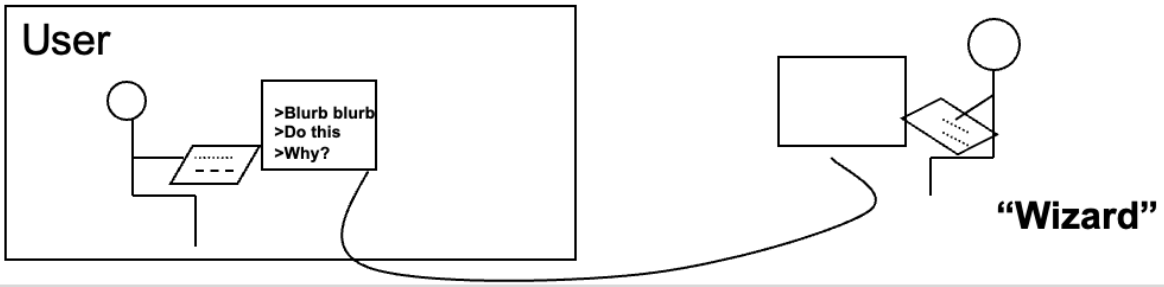
Wizard of Oz prototypes

- *"Pay no attention to the man behind the curtain"*
- Deceptively simple
- Appearance of a high degree of interactivity
- Highly flexible prototype
- Input actions sent to hidden person at evaluator's computer
- Sends appropriate simulated output back to user's computer
- Gives designers an idea of what shoulda/coulda been done by the UX design

“Wizard of Oz” prototypes Advantage

- Simulate behavior
  - In complex situations

  where user inputs are unpredictable.



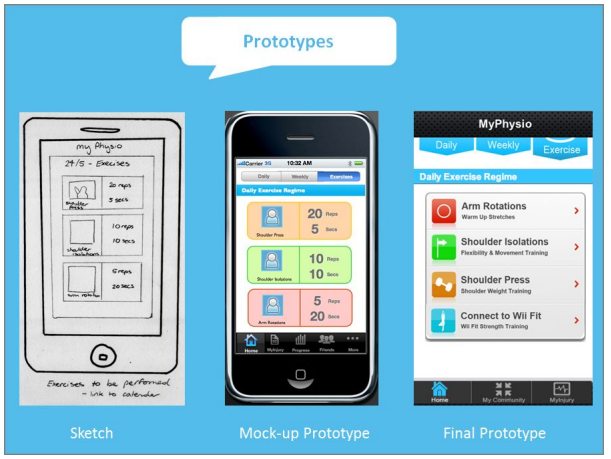
- Examples (in class)
- (demos that people don't always do what you expect!)
  - http://www.youtube.com/watch?v=ppnRQD06ggY&feature=related
  - http://www.youtube.com/watch?v=GrV2SZuRPv0&feature=related
- (good example of palm pilot prototype)
- http://www.youtube.com/watch?v=Bq1rkVTZLtU&feature=relate

Medium-fidelity prototypes

- Sometimes you'll hear about 'medium-fidelity' prototypes which are between low and high (but usually just low and high). This is a prototype with a level in between low and high fidelity
- Medium fidelity usually means more detailed wireframes (closer to final result but still no real functionality)
    - Good for intermediate design and early detailed design
    - To show
        - Layout
        - Breadth of user interface objects
        - Some work flow

High-fidelity prototypes

- After design is stable
- To demonstrate concept
- Include details of appearance and interaction behavior
- Required to evaluate design details
  - To get the design right
- How users can see complete (in sense of realism) design



High Fidelity Prototyping

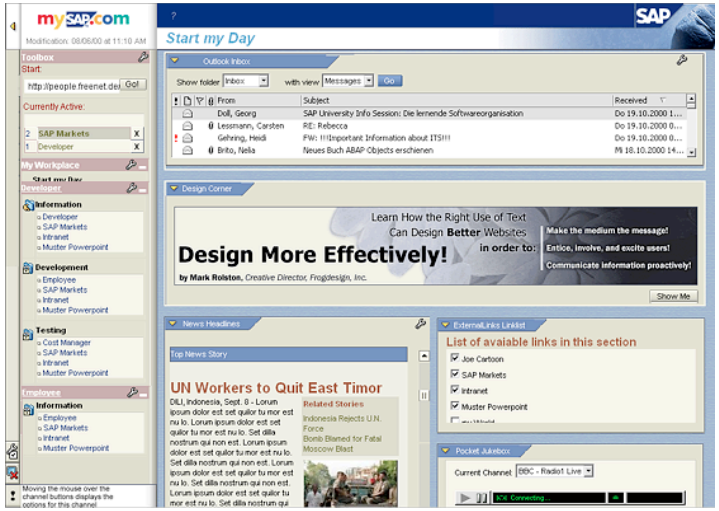
- Uses materials that you would expect in the final product and produces a prototype that looks much like the final version
  - –e.g., a prototype done in Visual Basic is a higher prototype than a paper prototype
- Uses software tools (e.g., Flash, Visual Basic, SmallTalk)

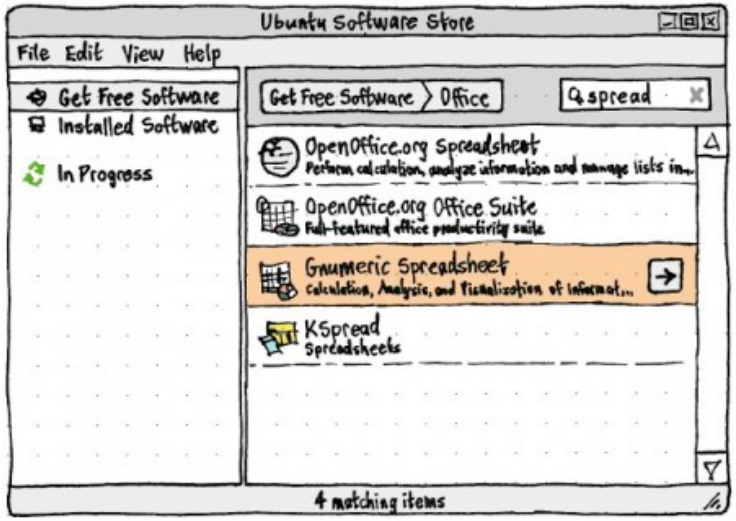
High-fidelity prototypes

- Still less expensive and faster than programming final product
- Useful as advance sales demos
  - For marketing
  - As demos for raising venture capital

- Problems with High Fidelity Prototypes
  - –Takes a while to build
  - –Reviewers and testers tend to comment on superficial aspects (e.g., the design) rather than the content
  - –Developers are reluctant to change something they worked on for hours

- High Fidelity Prototypes useful for selling ideas to people and for testing technical issues
- Low Fidelity Prototypes are useful for issues of content and structure





http://www.guuui.com/issues/03_05.php
http://andrewharvey4.wordpress.com/category/unswcourse/
http://www.sapdesignguild.org/resources/glossary_usab/index1.html

Fully programmed prototype

- Expensive, limited call for this (not used very often due to costs and because there are other ways of evaluating design and function with lower fidelity types).
- Good if you really need full-system operational prototype
- Rarely used

Prototype Example

- You are designing web site that allows users to order glasses.
- Scenario:
- This site will allow users them to try their glasses on virtually to find the best pair. Once they have found their perfect pair/s of glasses. They use the site to check out – including payment and shipping details.

- [can relate to Abby]. Abby needs new glasses. She uses the site to find new glasses after trying on several 'virtual' pairs. She then checks out and pays for her order.
- Possible Tasks:
  - Search frames
  - Try frames on
  - Check out for orders
  - etc.

Prototype Example

Setting: An online website to order eye glasses.

The User has selected the glasses they want and now need to to place their order

Normal Course:

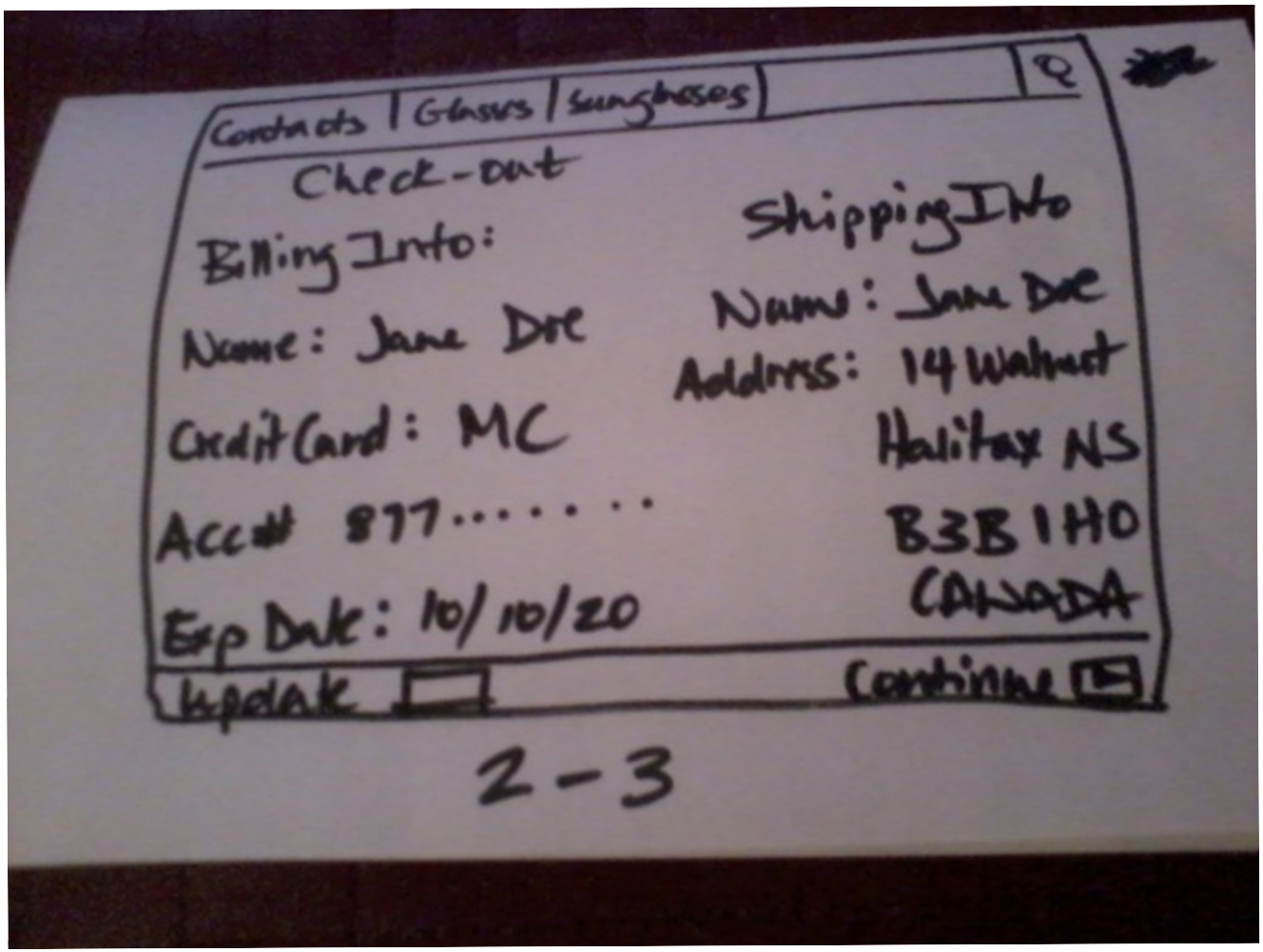
1. The user will indicate that she wants to order the items that have already been selected.
2. The system will present the billing and shipping information that the user previously stored.
3. The user will confirm that the existing billing and shipping information should be used for this order.
4. The system will present the amount that the order will cost, including applicable taxes and shipping charges.
5. The user will confirm that the order information is accurate.
6. The system will provide the user with a tracking ID for the order.
7. The system will submit the order to the *fulfillment system* for evaluation.
8. The *fulfillment system* will provide the system with an estimated delivery date.
9. The system will present the estimated delivery date to the user.
10. The user will indicate that the order should be placed.
11. The system will request that the *billing system* should charge the user for the order.
12. The *billing system* will confirm that the charge has been placed for the order.
13. The system will submit the order to the *fulfillment system* for processing.
14. The *fulfillment system* will confirm that the order is being processed.
15. The system will indicate to the user that the user has been charged for the order.
16. The system will indicate to the user that the order has been placed.
17. The user will exit the system.

Task Scenario

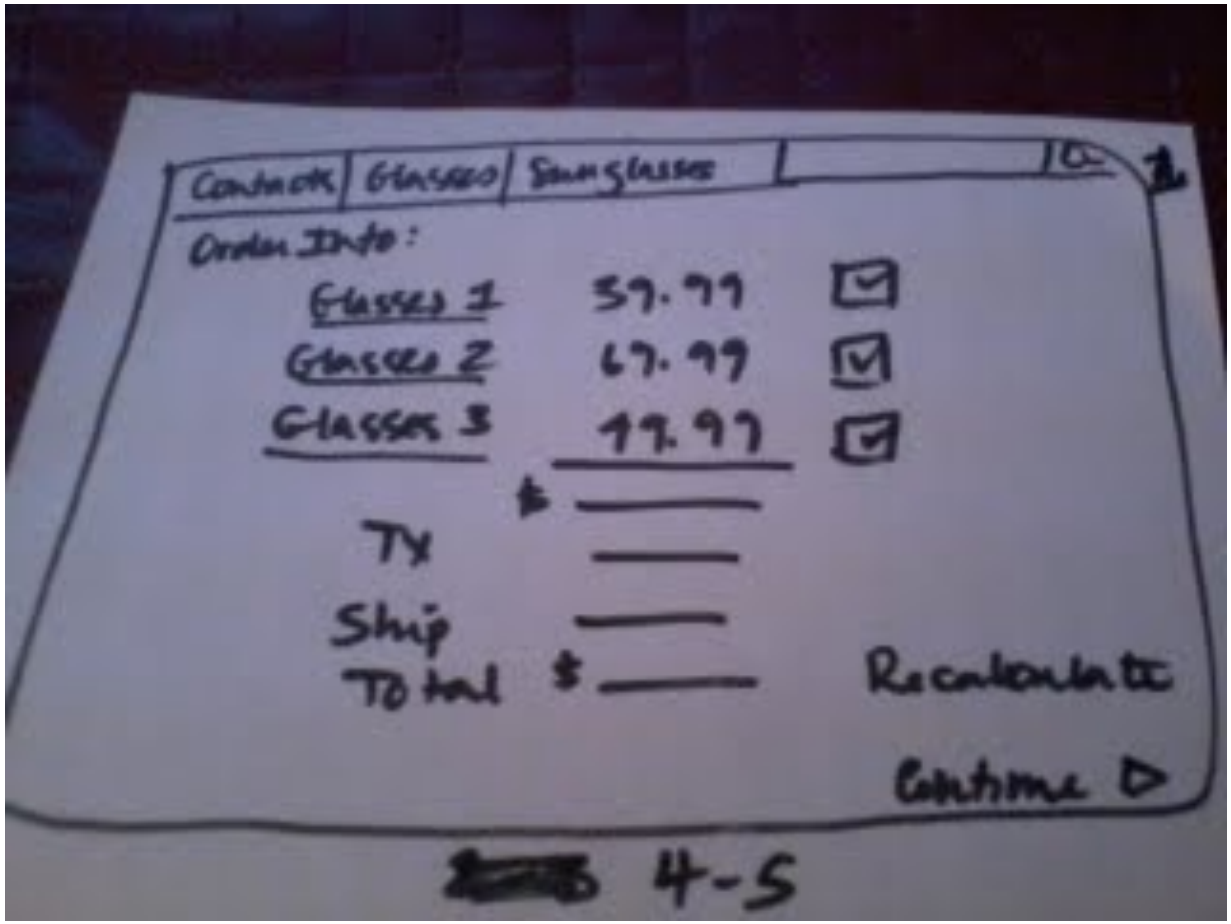
- Using these steps you can also create a scenario or task that you will ask the users to do (that way you can 'pre-program' the prototype so it looks like it's 'real'):
- For example – the following scenario:
- Abby, a 4th year student at Dal, has just selected three set of glasses (2 glasses and 1 pair of sunglasses). She now needs to checkout. She need to ensure that your previous shipping and billing info are still correct and find out about cost and shipping details.

1. Abby indicates that she wants to order the items that have already been selected.

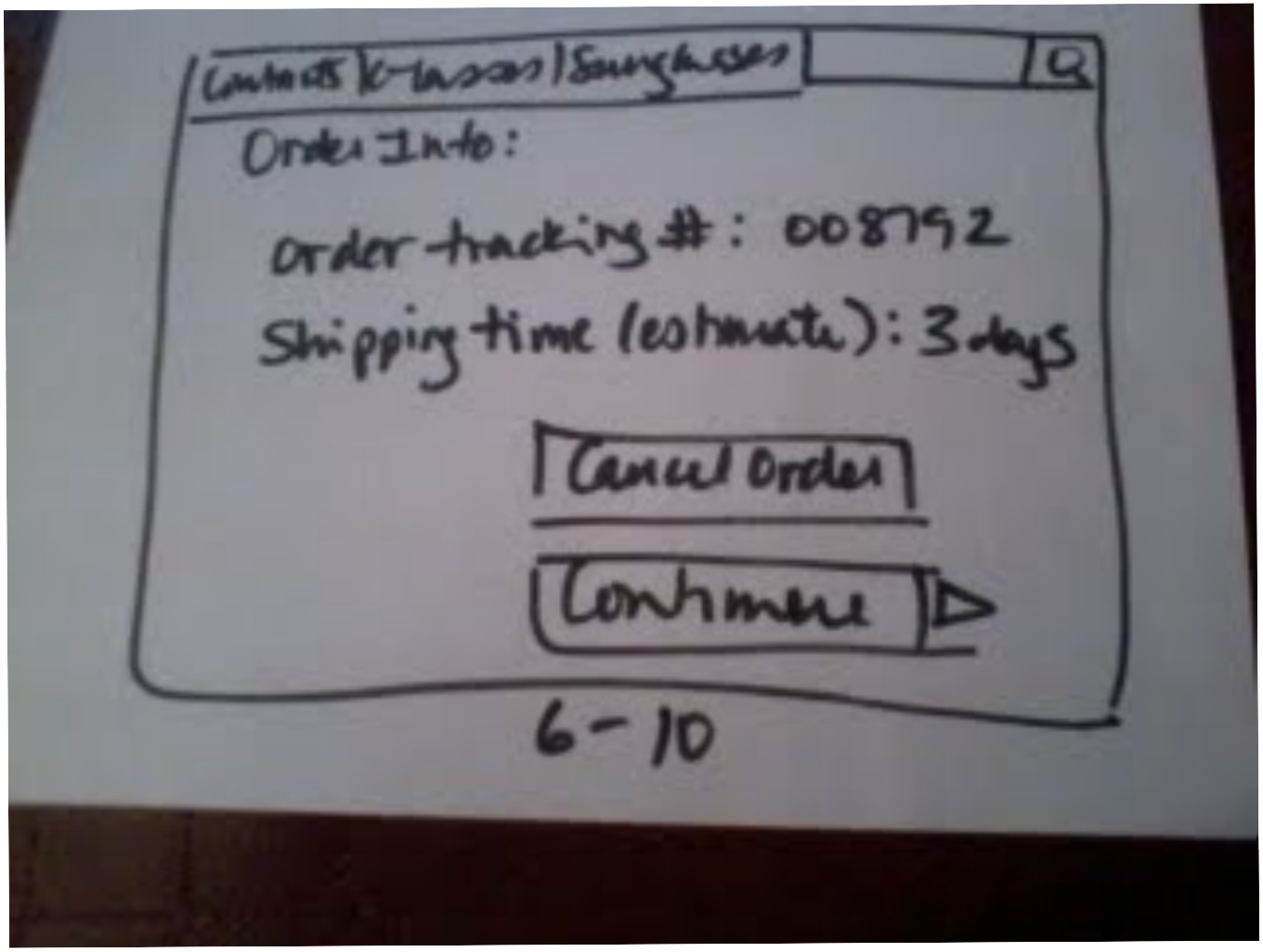
~~The user will indicate that she wants to order the items that have already been selected.~~



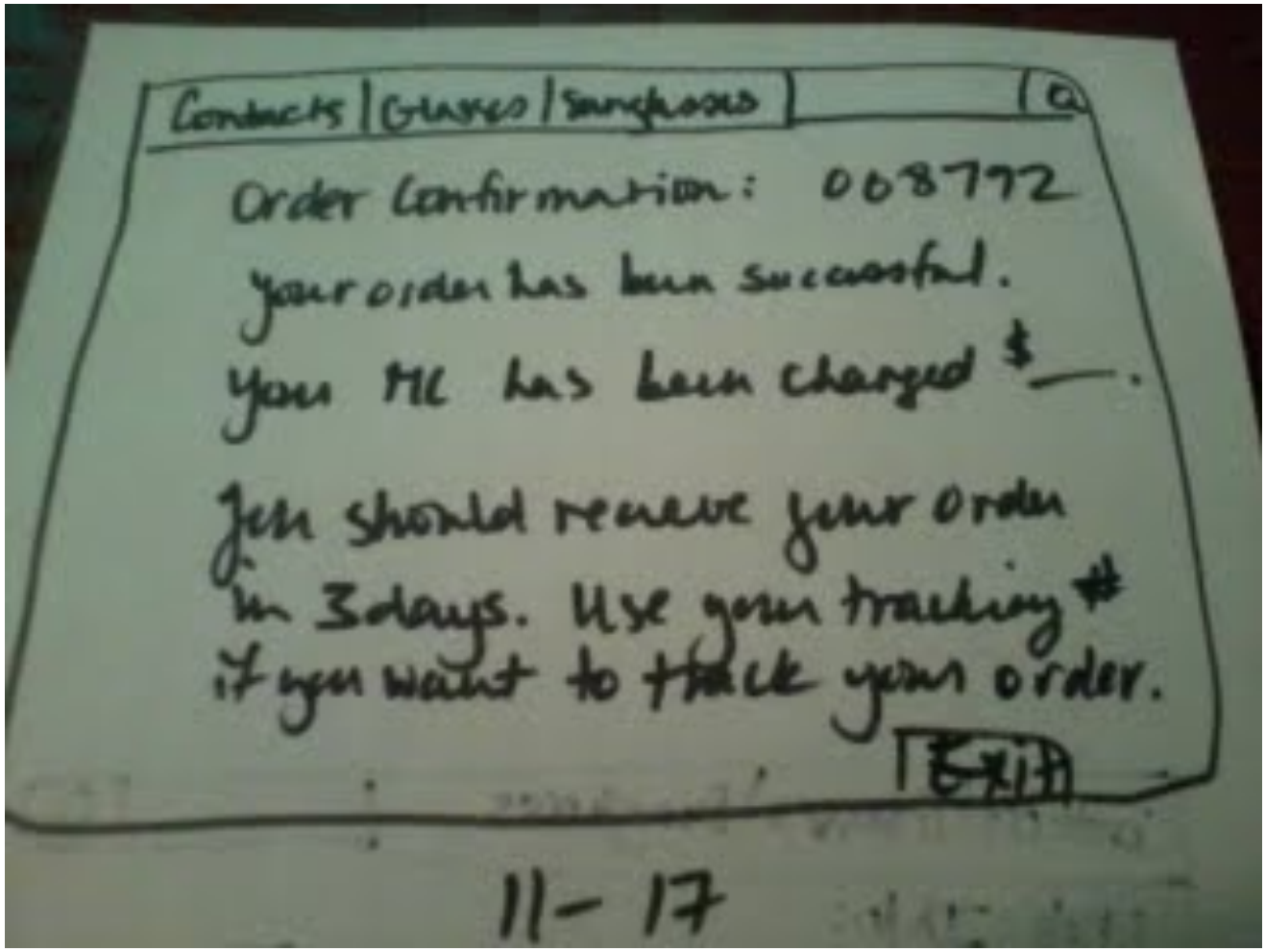
2. The system will present the billing and shipping information that Abby ~~the user~~ previously stored.
3. Abby ~~The user~~ will confirm that the existing billing and shipping information should be used for this order.



4. The system will present the amount that the order will cost, including applicable taxes and shipping charges.
5. Abby ~~The user~~ will confirm that the order information is accurate.



6. The system will provide Abby ~~the user~~ with a tracking ID for the order.
7. The system will submit the order to the *fulfillment system* for evaluation.
8. The *fulfillment system* will provide the system with an estimated delivery date.
9. The system will present the estimated delivery date to Abby. ~~the user.~~
10. Abby ~~The user~~ will indicate that the order should be placed.



11. The system will request that the *billing system* should charge Abby ~~the user~~ for the order.
12. The *billing system* will confirm that the charge has been placed for the order.
13. The system will submit the order to the *fulfillment system* for processing.
14. The *fulfillment system* will confirm that the order is being processed.
15. The system will indicate to **Abby** ~~the user~~ that the user has been charged for the order.
16. The system will indicate to **Abby** ~~the user~~ that the order has been placed.
17. Abby ~~The user~~ will exit the system.

Example 2: Automated Fish Feeder Application

- Design a touch-screen interface for an automated fish feeder device.
- Users can fill up the feeder before going on vacation and it will dispense food to their pet fish while they are away. The feeder will contain a weigh scale so that it always knows how many grams of food it contains.

Requirements

- It must support the following features and constraints:
- Users can specify when to feed the fish (specific times of the day or days of the week) and how many times per day or week to feed them (up to 3 times per day).
- Users can specify how much food to feed the fish at once (in grams).
- Users can specify feeding schedules and amounts for up to two types of food.
- Users can specify when long the automated program should run (start and end dates; including no end).
- The device will warn the user before confirming if there is not going to be enough food for the entire period.

Example Task:

- Set up the feeder for a set schedule (e.g., 2 weeks) for two different kinds of food.
- CREATE the USE-CASE

Task Scenario (**Use Abby from our persona**):

Feature1: feed fish for set time on a schedule
Task Name: Feed fish using the scheduler
Task Scenario: Abby is a Dal 4th year student who has a small aquarium with several fish. Abby is going home to Bridgewater over the winter break. She sets up her automated fish feeder to feed the fish twice per day for those 2 weeks, with flake food every morning at 9 am and pellet food every evening 6:00pm.

Use-Case

Normal Case

1. The system displays the main menu.
2. The user selects the option to set up a feeding schedule.
3. The system displays options for setting up dates for the feeding schedule: It asks the user to tell it the start date and end date.
4. The user enters the start and end date.
5. The system should check that the dates are valid.
6. The system then asks the user how many times a day they want to feed the fish.
7. The user enters in the daily feeding number.
8. The system prompts the user for the times for the daily feeding schedule.
9. The user enters the times.
10. The system asks user for which food to use (up to two choices).
11. The user enters food options.
12. The system for each food option asks for grams (amount) per feeding.
13. The user enters the food amounts.
14. The system checks amount with acceptable levels.
15. The system prompts the user to enter in food type for each time in the daily schedule.
16. The user enters the food for each time.
17. The systems display all the information.
18. The system prompts user to accept the information.
19. The user accepts and logs out.

Alternative Case

4.1 The dates are incorrect. The system asks the user to re-enter the date (#4)

Etc.

Prototype Images (you may have more or fewer images that rows – add and

Image 1: Steps 1 - 2

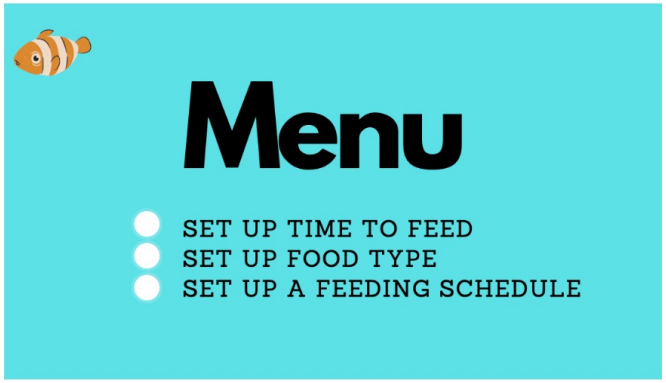


Image 2: Steps 3-5

Image 3: Steps 6-7

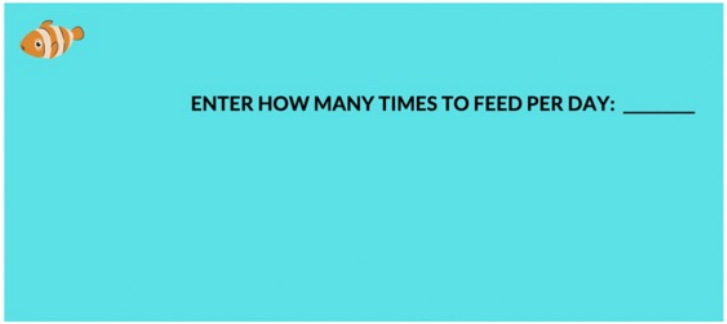


Image 4: Steps 8-9

Continue for all steps in the NORMAL use-case and for each feature.

Used https://www.canva.com to create this low fidelity prototype

Prototype Images (you may have more or fewer images that rows – add and

Image 1: Steps 1 - 2

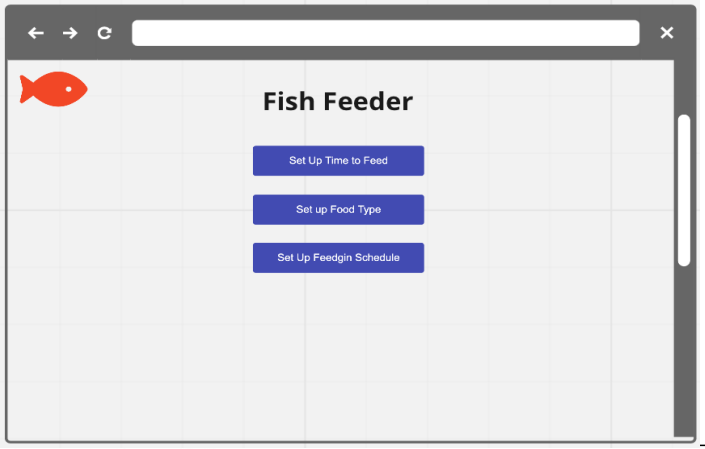


Image 2: Steps 3-5

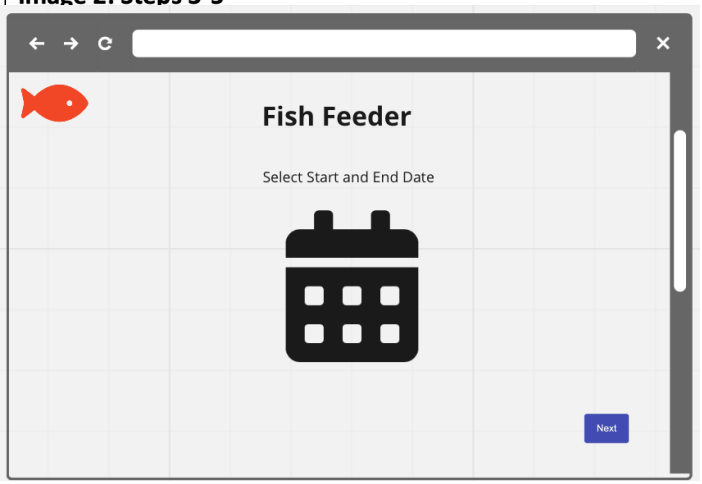


Image 3: Steps 6-7

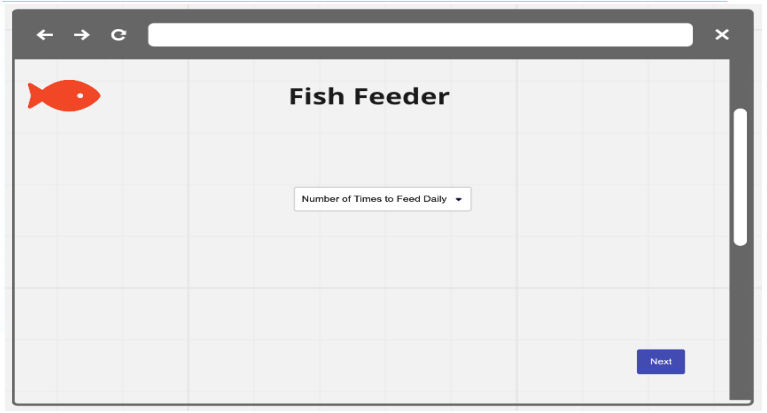


Image 4: Steps 8-9

Continue for all steps in the NORMAL use-case and for each feature.

Used https://www.miro.com to create this low fidelity prototype

- http://www.computerhistory.org/revolution/mobile-computing/18/341/1812

http://www.agile-ux.com/tag/paper-prototyping/
http://hci.stanford.edu/courses/cs247/2009/handouts/paper-2009-exercise.html
http://www.youtube.com/watch?v=LL0Y4u1UZGI&feature=fvsr
http://www.youtube.com/watch?v=GrV2SZuRPv0&feature=related
http://www.youtube.com/watch?v=U7U_zfGmSDk&feature=related
http://www.youtube.com/watch?v=gKkEmQjLYdg
http://www.youtube.com/watch?v=oITeUEjrY3Q&feature=related
http://tynerblain.com/blog/2007/04/09/sample-use-case-example/
http://samshoulders.com/tripassist.html
http://www.cs.washington.edu/education/courses/cse490f/07wi/project_files/camera2/c/proto/
http://www.benjaminlopez.net/recco.html
http://vis.berkeley.edu/courses/cs160-sp10/wiki/index.php/InteractivePrototype-Group:Epileptic_Eels