CS3205 – HCI in Software Development

Prototyping Strategies

Tom Horton

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

What Will We Be Talking About?

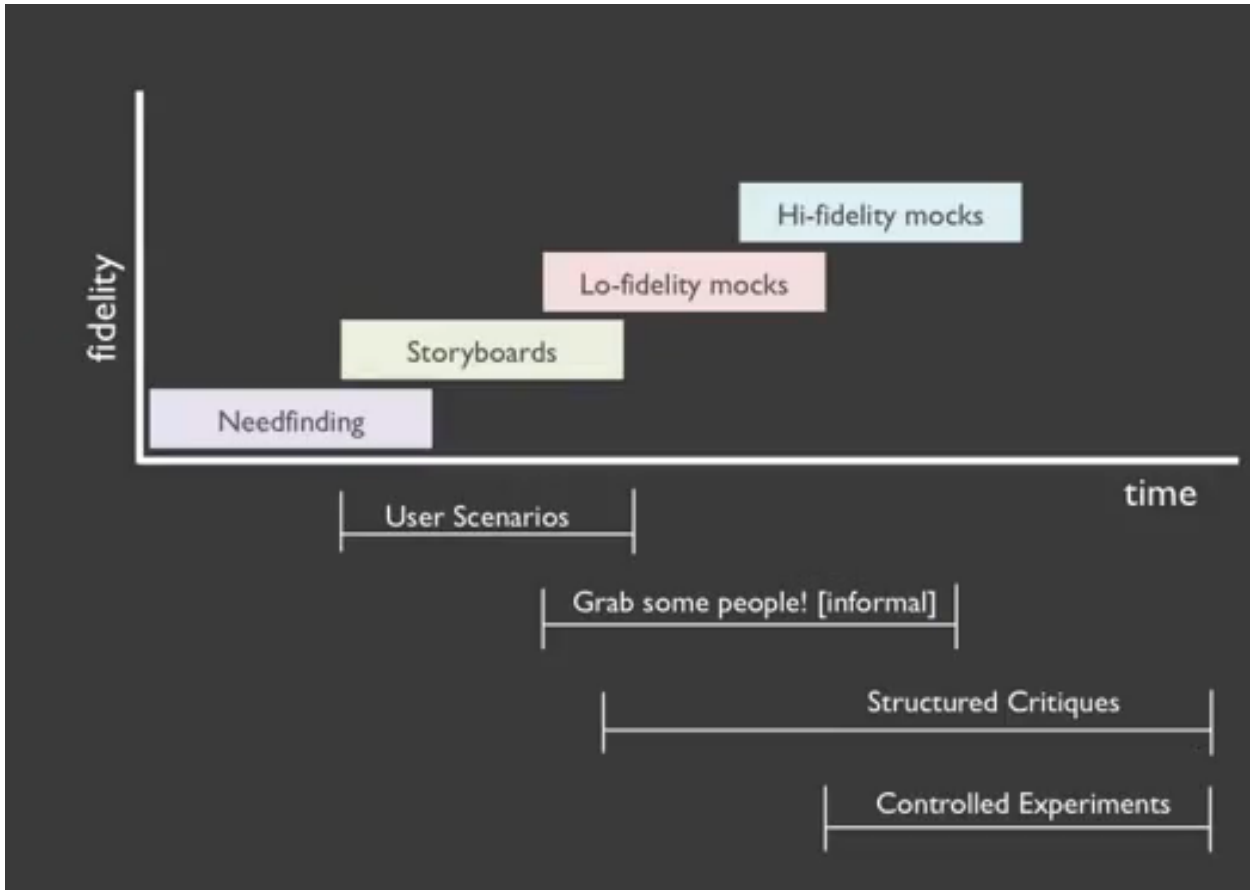
- Specific Prototyping Strategies!

- Low-Fidelity Strategies
  - Storyboarding
  - Video Prototypes
  - Paper Prototyping

Big Picture!

- Remember that you should always use stage appropriate tools!
  - Fidelity vs. Time graph! (next slide)
- The general rule is that you should progress from low to high fidelity slowly as the design process continues.

Prototyping Over Time

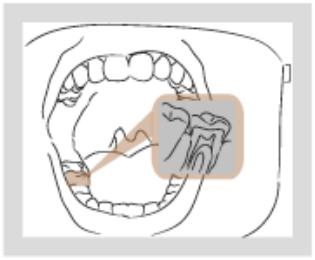


Storyboarding!

- Storyboarding is ALL about tasks!
  - NOT focused on specific interface elements and how they look, etc.
  - Thus it is incredibly low-fidelity.

Wearing special electronic glasses, the dentist examines the patient.

During the last visit, the dentist had noticed potential decay on tooth #29 and had marked it as a watch. The glasses now highlight this tooth, letting him know he should reexamine it.



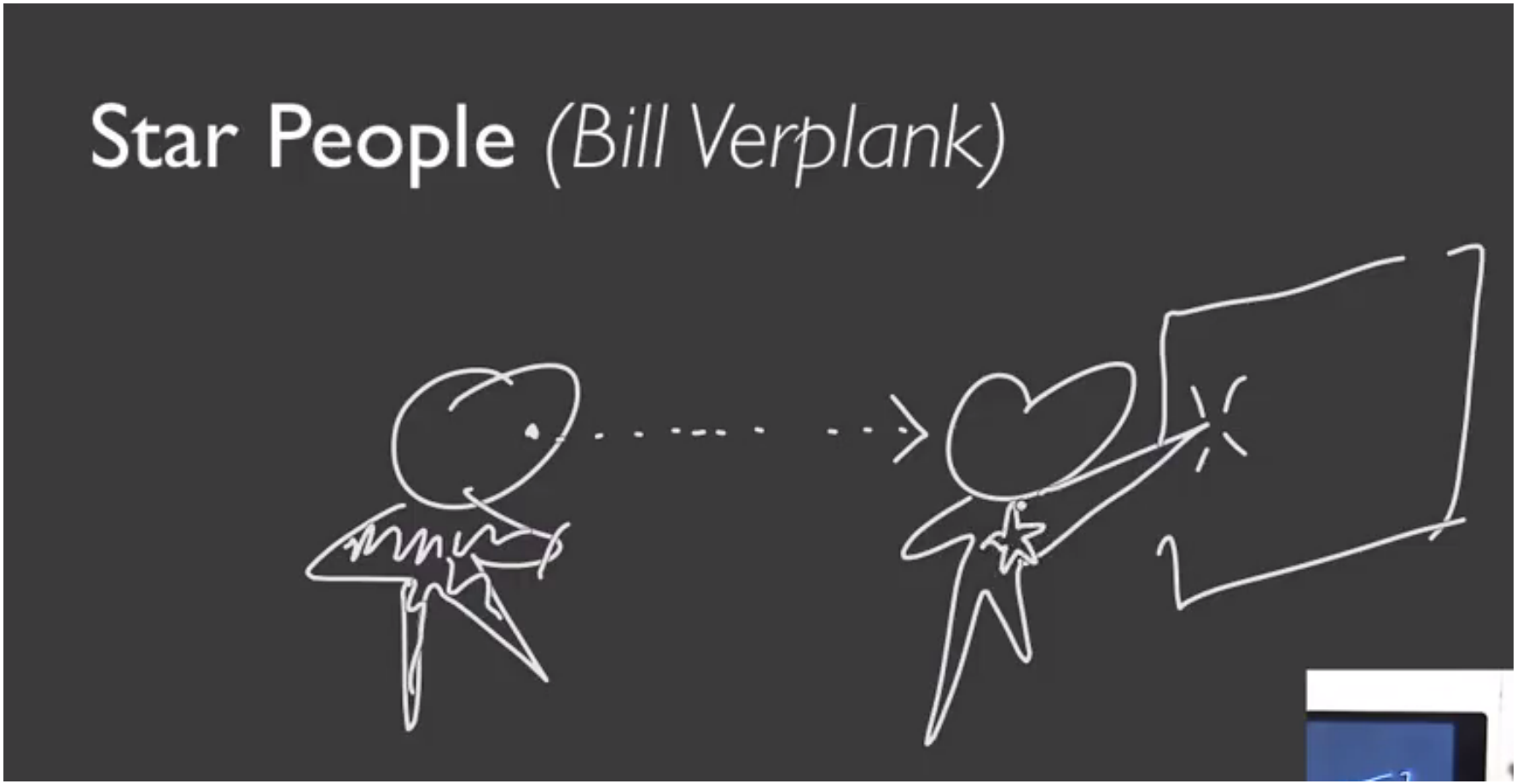
Focusing on the tooth brings up the radiograph of the tooth, all within his glasses.

Everything looks fine, and the dentist makes note of this progress.

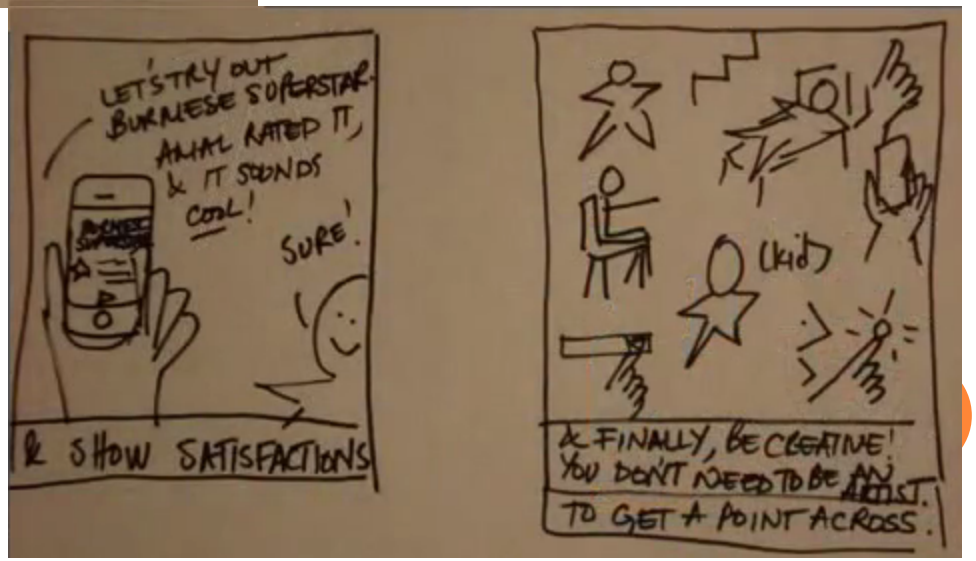
Storyboarding

- Storyboarding IS about
  - The tasks users want to complete.
  - The FLOW of the user's interaction.

- Storyboarding is NOT about
  - Pretty pictures (in fact bad drawing helps focus on task and not on imagery).
  - Showing specific interface elements.


Star People (Bill Verplank)

Example Storyboard



Storyboards Should Convey

- Setting
  - People Involved
  - Environment(s)
  - Task being accomplished
- Sequence
  - What steps are involved?
  - What leads someone to use the app or system?
  - What task is being illustrated?

Storyboards Should Convey

- Satisfaction
  - What motivates people to use the system?
  - What does it enable people to accomplish?
  - What need does system fill?

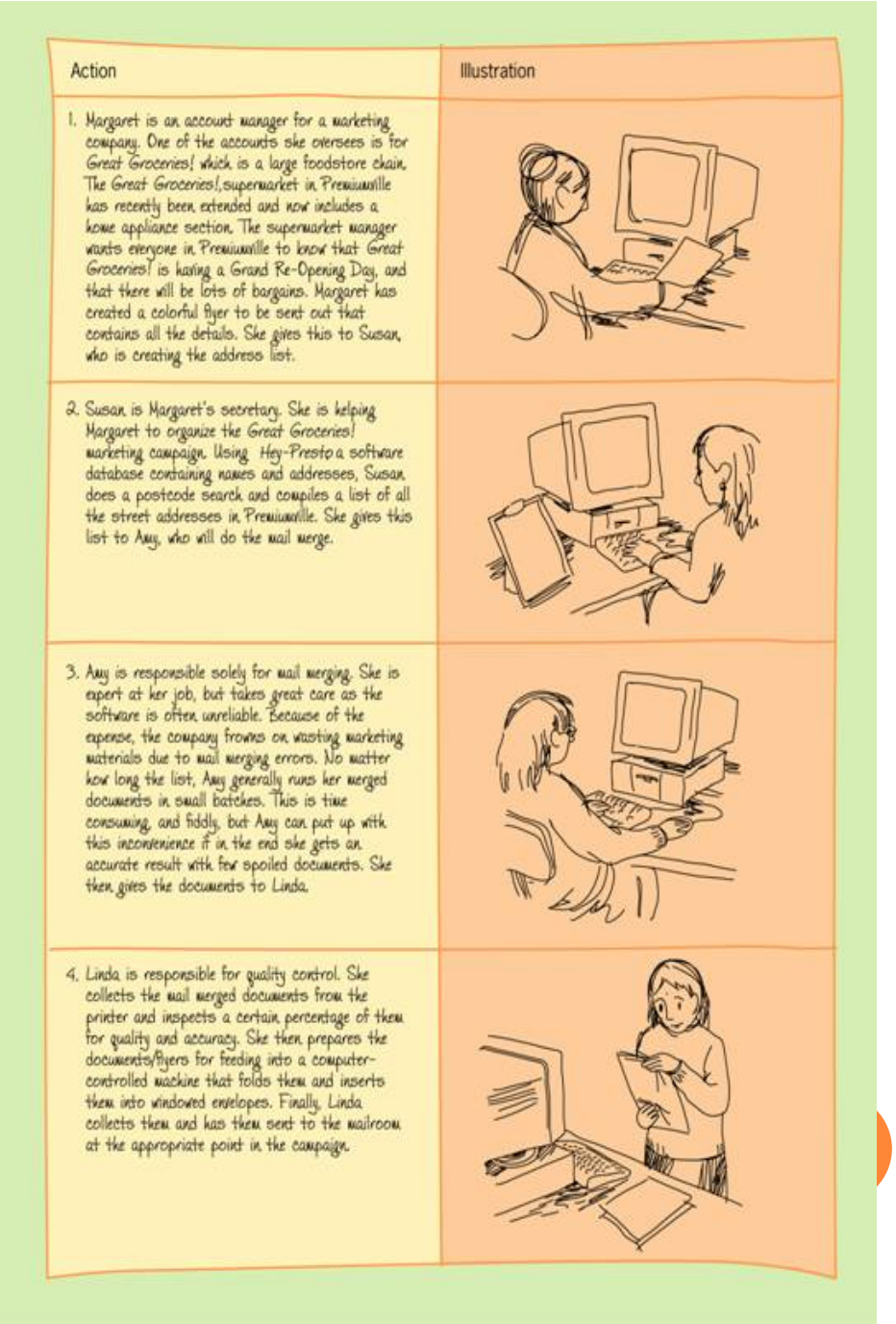
Benefits of Storyboarding

- Holistic Focus: helps emphasize how an interface accomplishes a task.
- Avoids commitment to a particular user interface (no buttons yet).
- Helps get ALL stakeholders on the same page in terms of the goal.

Some Advice Regarding Storyboarding

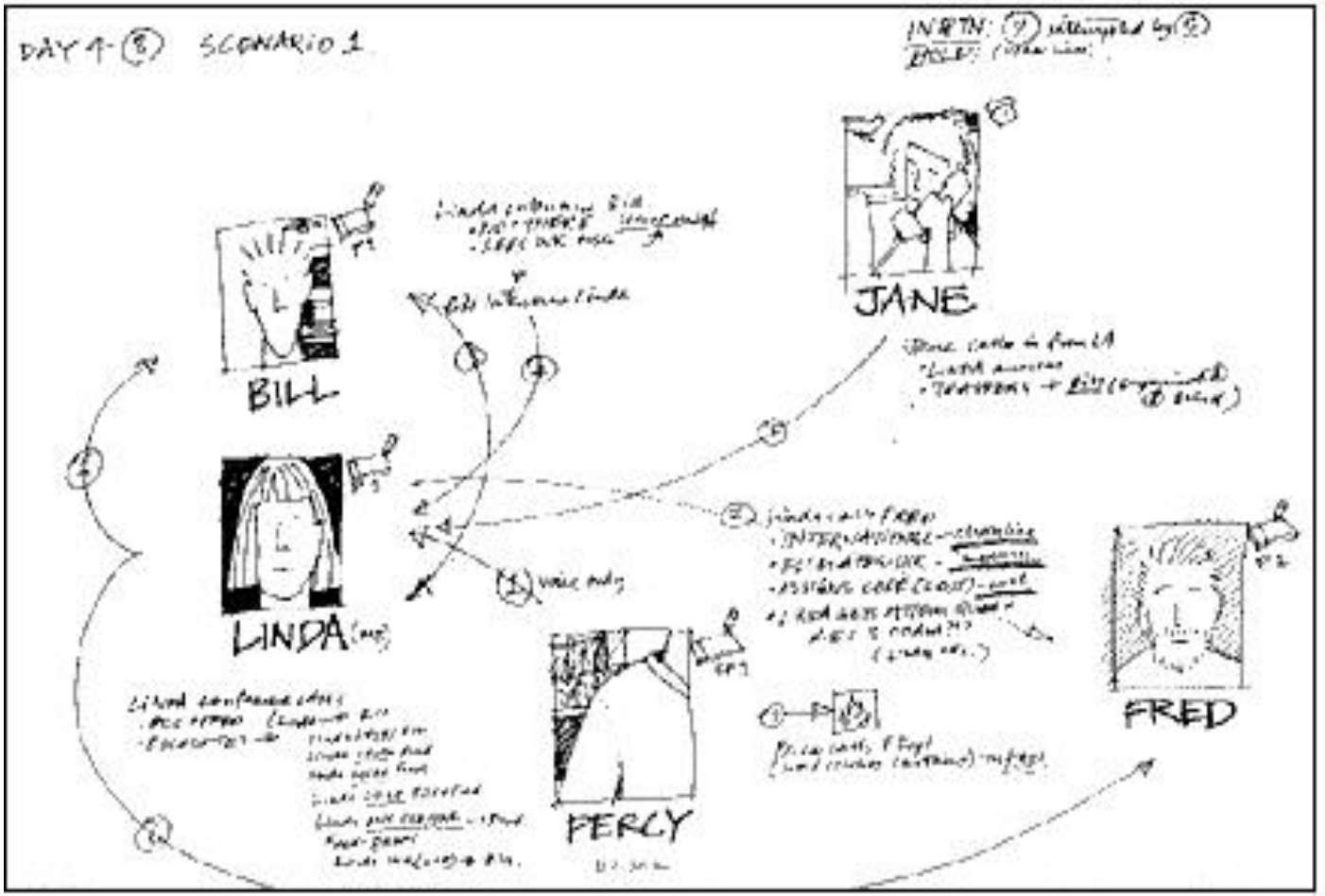
- DO NOT spend too much time on a storyboard.
  - Set a short time limit to enforce this.
- Focus on tasks and scenarios that are the MOST common for your users.

- Example from book *User Interface Design and Evaluation*, p. 119
- Shows
  - workflow of mail merging
  - who's involved, responsibilities, etc.



Action	Illustration
1. Margaret is an account manager for a marketing company. One of the accounts she oversees is for *Great Groceries!* which is a large foodstore chain. The *Great Groceries!* supermarket in Premiumville has recently been extended and now includes a home appliance section. The supermarket manager wants everyone in Premiumville to know that *Great Groceries!* is having a Grand Re-Opening Day, and that there will be lots of bargains. Margaret has created a colorful flyer to be sent out that contains all the details. She gives this to Susan, who is creating the address list.	
2. Susan is Margaret's secretary. She is helping Margaret to organize the *Great Groceries!* marketing campaign. Using *Hey-Presto* a software database containing names and addresses, Susan does a postcode search and compiles a list of all the street addresses in Premiumville. She gives this list to Amy, who will do the mail merge.	
3. Amy is responsible solely for mail merging. She is expert at her job, but takes great care as the software is often unreliable. Because of the expense, the company frowns on wasting marketing materials due to mail merging errors. No matter how long the list, Amy generally runs her merged documents in small batches. This is time consuming and fiddly, but Amy can put up with this inconvenience if in the end she gets an accurate result with few spoiled documents. She then gives the documents to Linda.	
4. Linda is responsible for quality control. She collects the mail merged documents from the printer and inspects a certain percentage of them for quality and accuracy. She then prepares the documents/flyers for feeding into a computer-controlled machine that folds them and inserts them into windowed envelopes. Finally, Linda collects them and has them sent to the mailroom at the appropriate point in the campaign.	

- This shows high-level of view of users involved in other storyboards



From: Usability Case Studies, http://ucs.ist.psu.edu

- Storyboard (or is it?) for a website
  - for photographers
- Sequence of pages
  - based on clicks
- Explanations / annotations

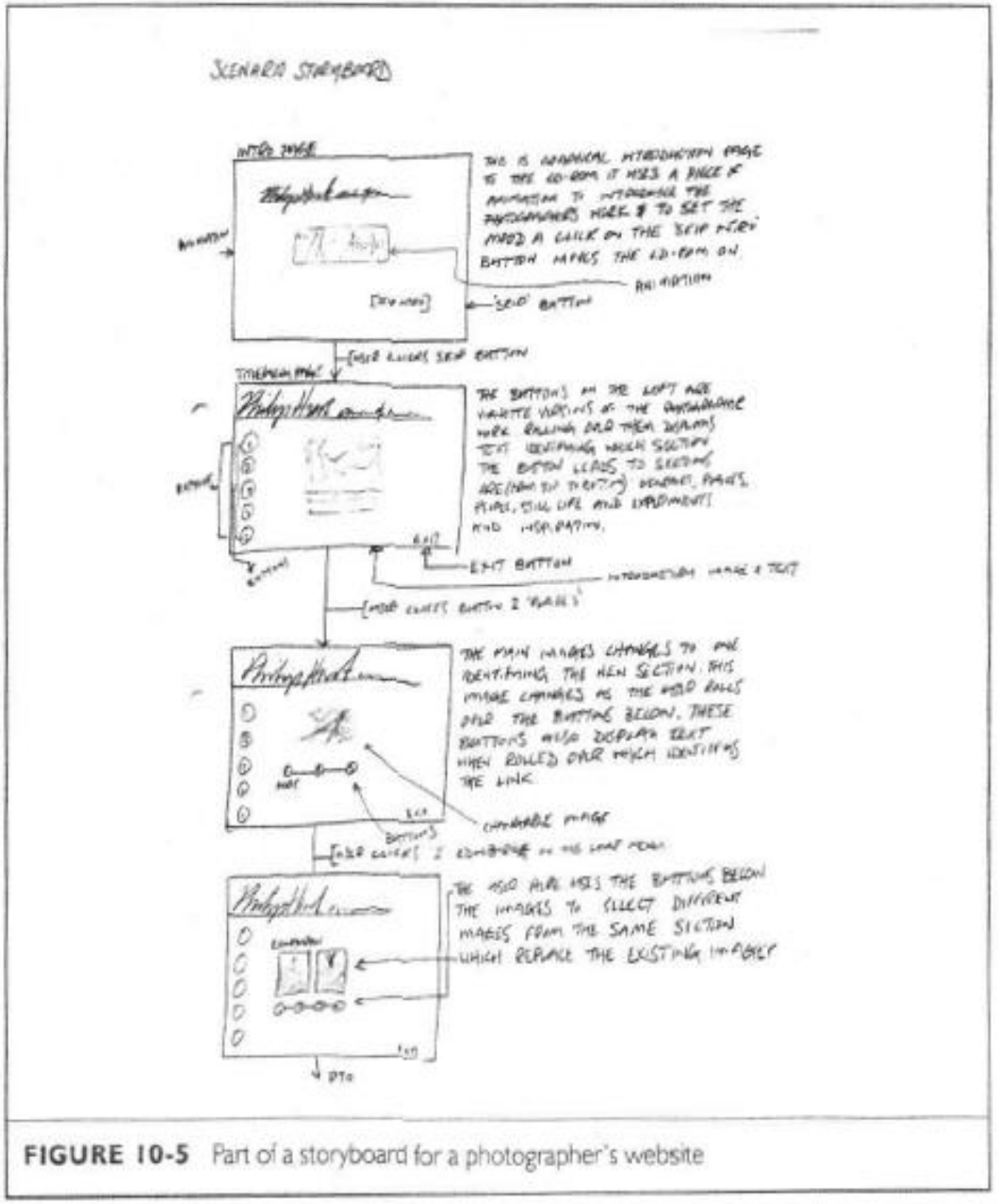


FIGURE 10-5 Part of a storyboard for a photographer's website

From book: *Designing Interactive Systems*, 2005

Video Prototyping

- What is video prototyping?

Benefits of Video Prototyping

- Cheap and fast
- Great communication tools
  - Helps achieve common ground
  - Ideally, portable and self-explanatory
- Can serve as a ‘spec’ for developers
- Ties interface designs to tasks
  - Aligns and orients interface choices
  - Makes sure you have a complete interface

Video Prototyping

- Can be any level of fidelity.
  - Can make videos with paper prototype examples.
  - Fancy high fidelity videos are ok too!
- Usually, dependent on where you are in the design process.

What do you put in video prototype?

- The WHOLE task including motivation and success.
  - Establishing shots and narrative help.
- Draw on tasks you observe.
- Illustrate important tasks your system enables.
  - Not in video, probably not necessary for version one of the system.
- Can help scope a minimum viable product.
- Changes what design teams argue about (in a good way).

Steps to Video Prototyping

- 1) Outline / Storyboard the Video
- 2) Obtain Equipment.
  - A camera (nothing fancy required). Could be a phone even!
  - People
  - Realistic Location
- 3) Focus on MESSAGE, not on production value.
- 4) Film it!!

Video Prototyping Examples!

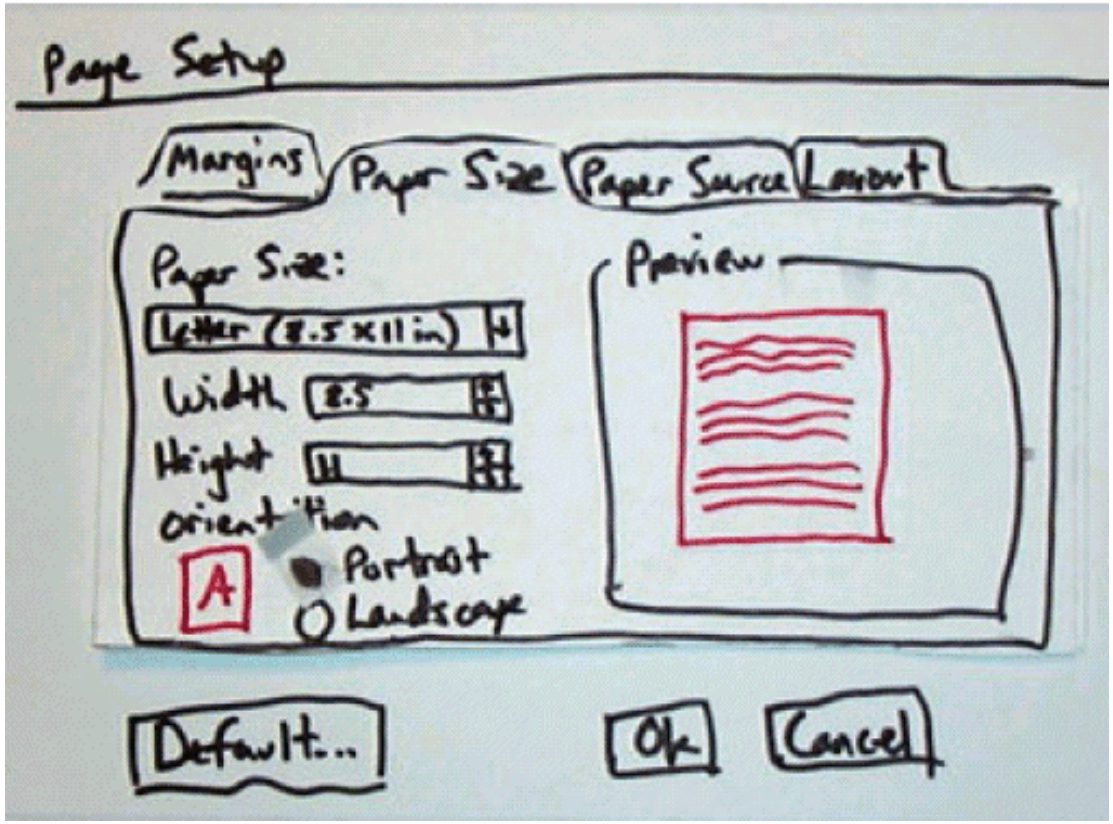
- https://www.youtube.com/watch?v=kWsBvUnvCmg (shown in class)
- https://www.youtube.com/watch?v=fpHm0dCHxpk
- http://www.youtube.com/watch?v=OlQvolZEtcE
- http://youtu.be/UAOptso2owU

Considerations

- Audio? Not always necessary.
  - You can verbally describe video in person while showing it to someone.
- Interface fidelity? Paper, digital, etc.
  - Can even be invisible!
- Show both success and failure. Several use-cases always a plus.
- Don't spend too much time editing.

Paper Prototyping!

- Quite literally, using paper to produce a potential interface design. (Cardboard, post-its, tape, index cards, glue-sticks, etc.)



Benefits of Paper Prototyping

- Very Fast! Very Cheap!

- Easy to change and adapt
  - Why not code-it, real GUIs, using a builder tool, etc.?
    - (I.e. high-fidelity)
  - Estimated to be 100 times cheaper if NO code is written. (So says Jakob Nielsen, so you know it's true.)
  - Can be used to collect important usability information!

- Can support interaction: an evaluation session with users
  - Really? But it's just paper, scribbling and post-it notes!

User Interaction with a Paper Prototype

- Evaluation session

- What happens:
  - User walks through a task
  - One team member plays computer
  - New screens etc. are moved in and out
  - Team draws on paper, adds things, moves things, etc.
  - Someone ask questions, elicits comments
  - Some records feedback (video, notes)

- Benefits!
  - Not tempted to focus on fit-and-finish
  - Allows focus on a desired evaluation goals (e.g. flow, conceptual model, visibility (?), etc.)
  - No “demo fail”. Easy to change on the fly during evaluation session.

Additional Advantage: Collaborative Design

- Users (and Stakeholders) can get involved and quickly alter / update the interface.

- Often this leads to a better design because they can show you what they prefer!
  - Scaffold their efforts.

Demonstration Video

- Hanmail: https://www.youtube.com/watch?v=GrV2SZuRPv0
- Coloring App for Kids (good evaluation session example too): https://www.youtube.com/watch?v=9wQkLthhHKA
- Corel's video on WebPoint product: http://www.youtube.com/watch?v=ppnRQD06ggY
- iPhone (color app, presentation by developer): http://www.youtube.com/watch?v=V8LNDqMIapY

Paper Prototyping Tips and Tricks

- 1) Keep materials in one place! Small interface widgets tend to get lost or damaged easily.
- 2) Work quickly and make reusable components.
  - Screen grab UI widgets. Photocopy things. Use scissors (but don't run with them).
- 3) If something is difficult to simulate (progress indicators, right mouse menus, hyperlinks), have the user ask if it is available and then verbally describe the interaction.
  - E.g. you don't have to construct (or code) everything

Paper Prototyping Tips and Tricks

- 4) Backgrounds (Poster Board, etc.) can be useful to contain the prototype and provide context for the user.
  - E.g. a iPad or iPhone template
- 5) Don't be afraid to mix and match hardware and software! Maybe a physical block of wood with paper on it!
- 6) When appropriate, add context by including familiar operating system elements.

Paper Prototyping Materials

- Widgets!

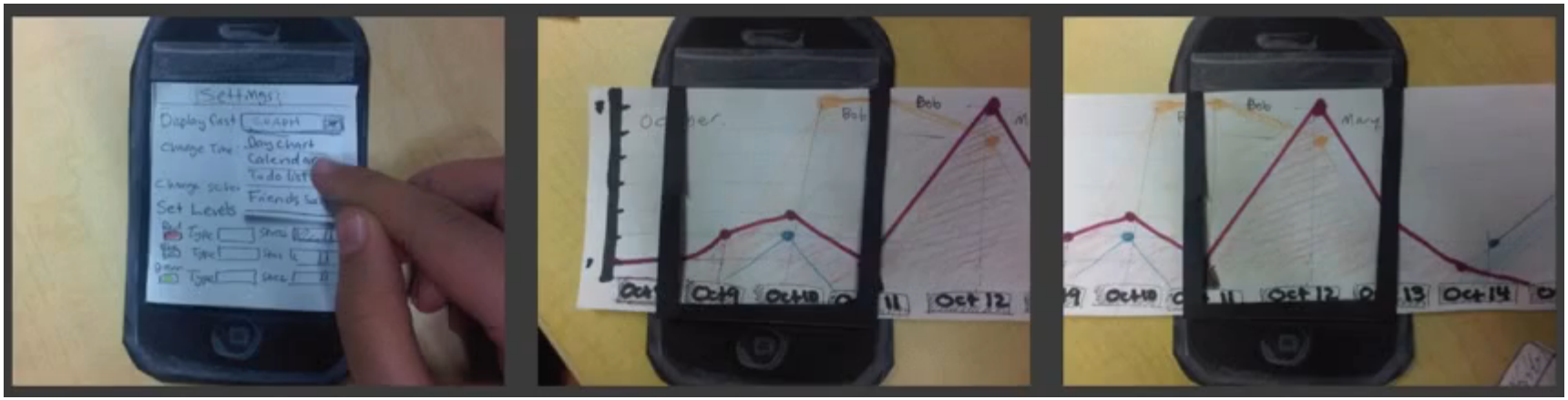
Paper Prototyping Materials

- Connectors

Paper Prototyping Materials

- Drawing

Another Example

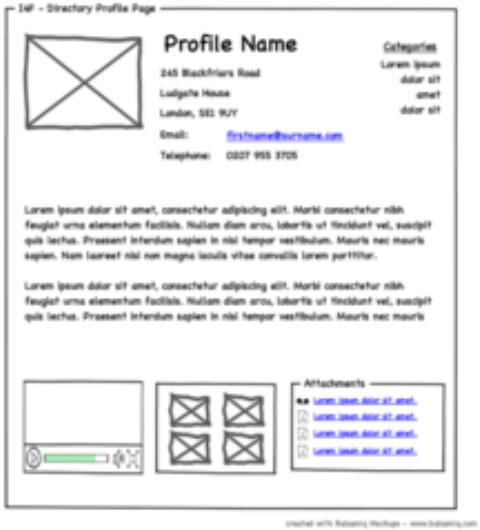


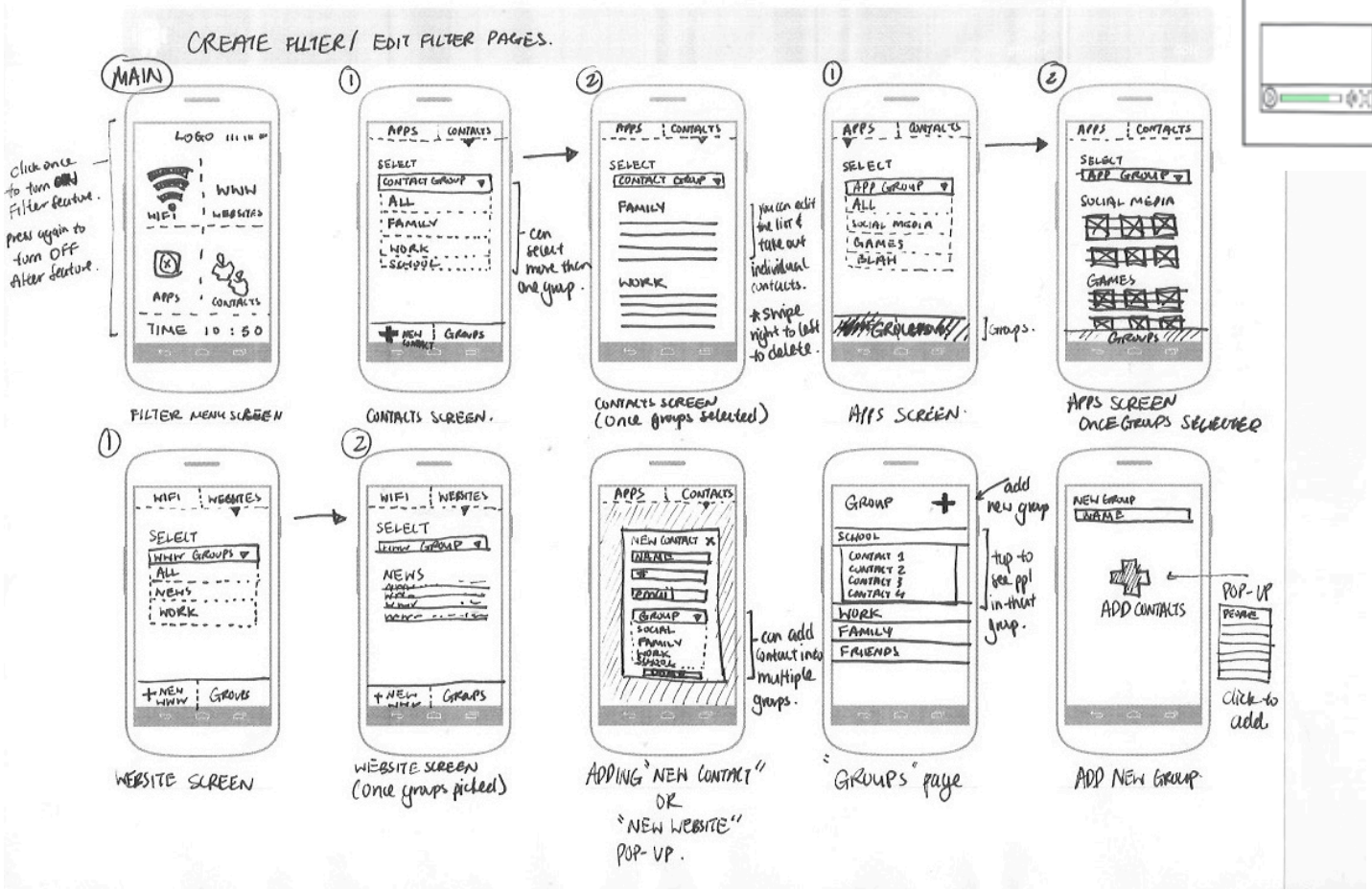
Wireframing

What's a Wire Frame?

- A low fidelity representation of a single page or screen
- Goal: to meet the purpose of that screen, what UI elements are needed, and how are they organized?
- Should show:
  - All content or features needed
  - Relative sizes and locations of these
- Should NOT include:
  - Colors
  - Fonts
  - (Other "fit and finish" characteristics)

EXAMPLES





- https://en.wikipedia.org/wiki/Website_wireframe
- http://anythingnet.com.au/wp-content/uploads/2015/05/app-development-wireframing-flowchart.jpg

Role of Wireframes

- When in development?
  - After finding user analysis, understand requirements.
  - Perhaps after paper prototypes
  - As a step towards designing the final interface
- Keep a low-fidelity approach
  - Colors, fonts, etc. – people begin to focus on fit and finish, and not on what you want
    - Which is: can this do what we need it to do? Verify against…
- Should reflect what you learned about users, scenarios, etc.

Wireframing Tools

- Many! Some are web-based, some are apps
- Balsamiq
  - Draws things to make things look like paper (lo-fi)
- Other drawing tools
  - e.g. Microsoft Visio, OmniGraffle
- Others
  - Pencil Project (free)
  - Sketch
  - Marvel
  - MockFlow
  - FluidUI

Wireframe vs. Paper Prototypes

- Wireframes are
  - More complete
  - More polished (still lo-fi)
  - Often created with computer tools
- Paper prototypes
  - Done earlier, perhaps
  - Less polished (perhaps)
  - Seem to be used more for evaluating interaction
    - Doing an evaluation with users
- Perhaps one thing they both don't show well is…

Overall UI Flow

- Major UI elements are probably windows or screens (on handhelds)
  - What are they? How are they organized?
  - Do they "match" users' mental models of how they want to achieve tasks?
  - Compare to SW architecture design
    - Major components and their roles
    - How they're related
    - Save detailed design (inside modules) for later

Goal of Modeling UI Flow

- Identify major UI components (windows etc)
  - High-level description of their role
- Interactions between these
  - Conditions?

- Two levels possible
  - At conceptual design (more abstract)
  - At physical design (what will be windows/screens/pages)

UIDE book's Content Diagrams

- Content diagram is:
  - lo-fi prototype that shows organization and structure of the UI
  - from the designer's perspective (?)
- In physical design, can be transformed into GUI or website or…
  - May not be a one-to-one match between this level of abstraction and final physical design

Creating a Content Diagram

- Need to know:
  - Primary tasks, objects, etc. for this UI
  - Probably want to have use-cases, scenarios, etc.
- Then,
  - Identify **containers** and “task objects”
  - Decide which tasks go into each container
  - Define **links** that show navigation flow

Content Diagrams: Containers

- Container
  - an abstract representation of some part of a user's work activity
  - includes functions required to do that activity
- Various levels within the diagram
  - Main container should record
    - Vital tasks and Frequent tasks
  - "Lower-level" containers

Content Diagrams: Links

- Links
  - how the user will navigate in the UI between these functional areas
- Single link: next container becomes primary focus
- Double link: second container requires context of first container
  - Example: spell checking window in a word-processor
- Links may have labels to indicate conditions on them

Representing Containers

Name

Purpose

Functions

- {performed by the user}
- {performed by the computer system}

Links

- ▶ {single link}
- ▶▶ {double link}

Objects

Constraints

UIDE book's template

Note: Objects are Task Objects

Example Container

View search results

Displays the search results

Functions

- Show search results
- Select search results

Links

- View details

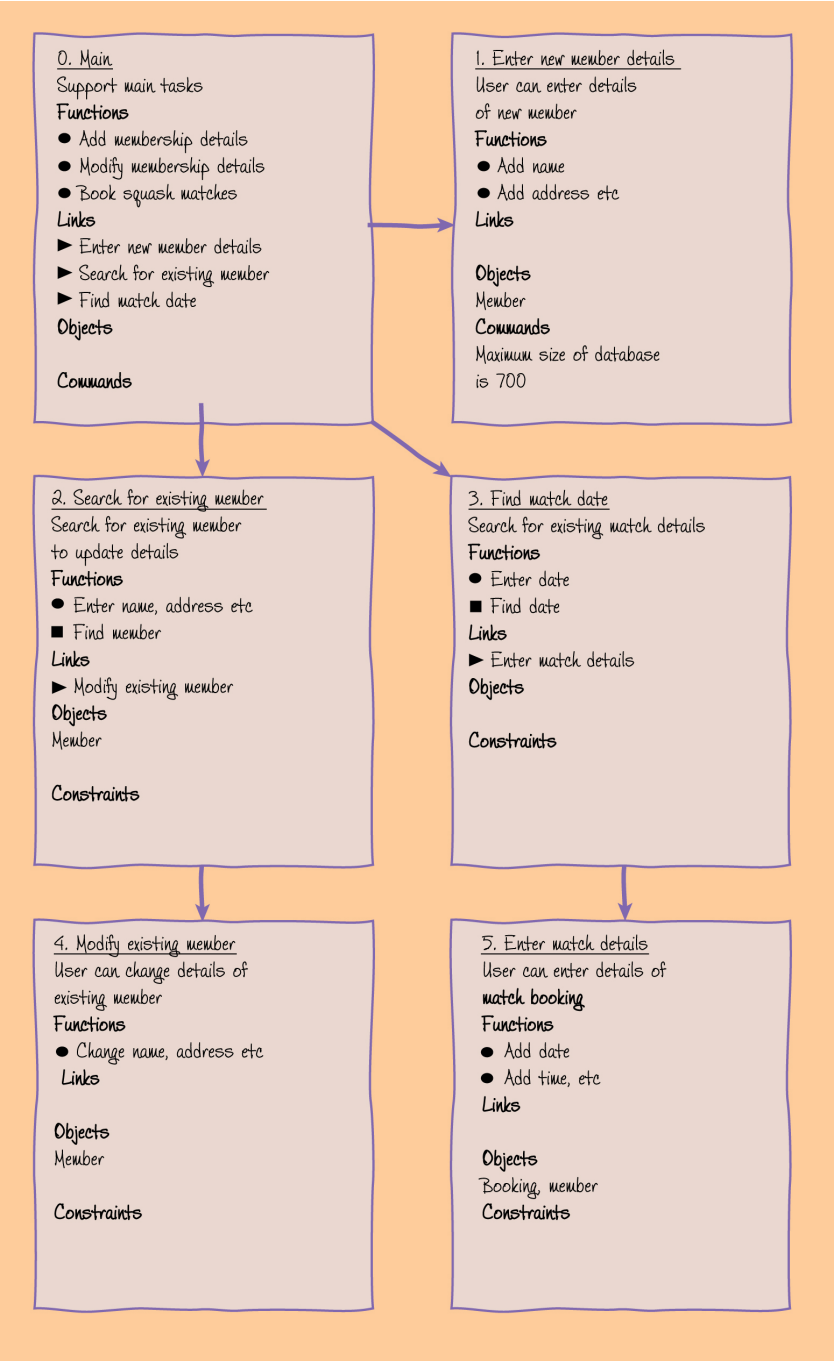
Objects

Resource

Constraints

Must be able to show at least five results at the same time

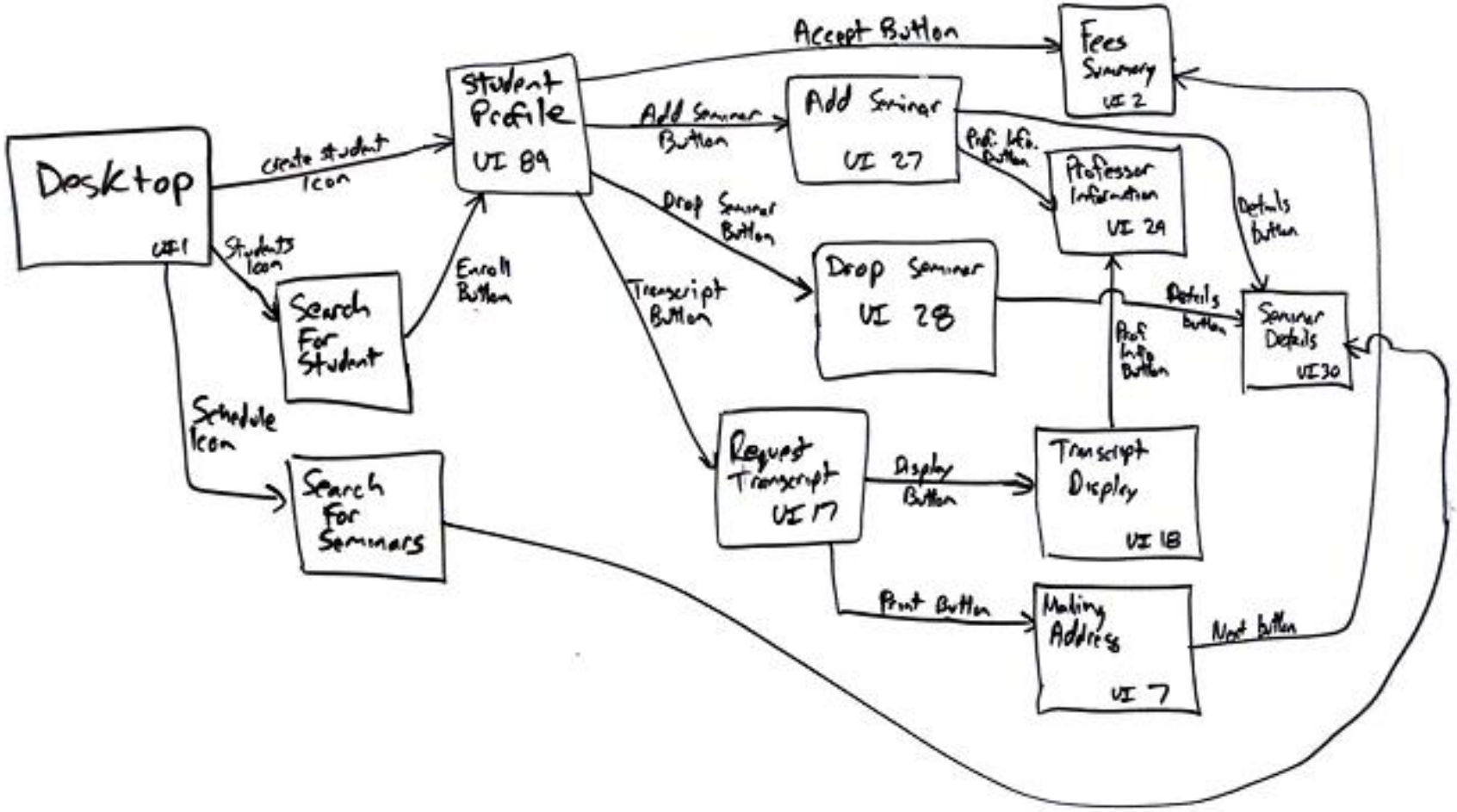
Example Content Diagram



Another Approach: UI Flow Diagrams

- We can do less than this and still benefit
- Prototype often called a **UI Flow Diagram** or **Navigation Diagram** in the Unified Process methodology
  - Post-its or boxes represent major UI elements (e.g. Windows)
  - Each has a title (or short description) and often an ID
    - ID used to refer to a more detailed diagram (perhaps done later)
- Goals:
  - Get high-level overview
  - Trace through use-cases or scenarios

Example: UI Flow Diagram



- From: http://www.agilemodeling.com/artifacts/uiFlowDiagram.htm

Web Design: Sitemaps

- (Note: We don't mean an index-like web page that users look at to understand what's on a website)
- A design representation for identifying the structure and high-level content of a website
  - What are the pages?
  - How are they linked?
- For planning and design, we need to know:
  - Goal or purpose of each page
  - What can we learn from this?
- Tools? Yes. Or, PostIt Notes on whiteboards