Design Process

SWE 795, Spring 2017

Software Engineering Environments

GEORGE MASON UNIVERSITY

Today

- Part 1 (Lecture)(~45 mins)
  - How do you conduct a think-aloud usability study?
  - How do sketching and prototyping fit into the design process?
- Part 2 (In-Class Activity)(~30 mins)
  - Conducting a think-aloud usability study
- Break!

- Part 3 (Presentations)(30 mins)
  - Project proposals
- Part 4 (Discussion)(45 mins)
  - Discussion of readings

Design Process

Exploratory studies

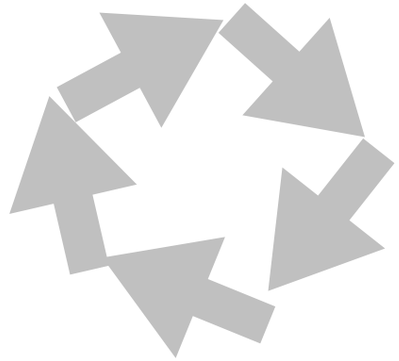
survey
indirect observation
contextual inquiry
...

Models

questions
information needs
use of time
....

Generate tool designs

scenarios
mockups



(Expensive) evaluation studies

lab study
field deployment

Implement tool

(Cheap) evaluation studies

heuristic evaluation
paper prototypes
participatory design
...

(Some) types of exploratory studies

- Field observations / ethnography / lab observations
  Observe developers at work
- Surveys
  Ask **many** developers specific questions
- Interviews
  Ask a **few** developers **open-ended** questions
- Contextual inquiry
  Ask **questions** while developers do work
- Indirect observations (artifact studies)
  Study artifacts (e.g., code, code history, bugs, emails, ...)

Observational Lab Study

Observational Lab Study

- Formulate **goals** of study
  - What are you trying to learn?
- Design study protocol, tasks, materials, data collection, …
  - Pilot study design
- **Conduct** study
- **Analyze** data to assess task performance and identify usability issues

Study goals

- What challenge about software development are you trying to understand?
  - Identify steps that are time consuming
  - Identify barriers that prevent developers from making progress
  - Identify breakdowns, where developers' mental model diverges from system (e.g., inserting defects)

Selecting participant population

- Who will be the users?
- Goal: users representative of system's **target users**
- Are there multiple **classes** of users (e.g., developers that know codebase well, developers new to project)?
  - If so, which are appropriate given goals?
  - May choose several classes

Number of participants

- More participants —> different participant interactions, more data
- Fewer participants —> faster, cheaper
- No right answer, as depends on potential diversity of interactions and users
  - Nielsen & Morlich (1990) found that 80% of problems could be detected w/ **4-5** participants
    - Most serious usually detected with first few

Consent

- Important for participants to be told up front what they will do and provide affirmative consent
- Helps allay potential participant fears
- Make clear purpose of study
- Make clear that you are evaluating your design, **not** the user

Tasks

- What will users do?
- Goals for task design:
  - Provide specific goal: something that the user should accomplish
  - Comprehensive enough to exercise key features of your app
  - Short enough to minimize participant time commitments

Communicating tasks

- Provide a scenario explaining the background of what users will be doing
- Provide a specific goal that the user should accomplish
  - But **not** how they should accomplish it
  - Don't give away how you hope users will accomplish goal
- Communicate **end criterion** for task - how do they know they're done?
- Provide maximum time limit after which they will be stopped

Recruiting participants

- Many potential sources
  - Co-workers, colleagues, friends, family
  - Email, mailing lists, online forums
  - Announcement at related user groups
- Important to select sources that best match the background & knowledge of target users

Training

- Goal: **avoid** unless really necessary
- Training necessary when
  - Participants require specialized knowledge to act as target users
  - Target users will have access to specialized training materials before they begin study

Data collection

- Think aloud
- Screencast
- Questionnaires or interview questions to gather participant feedback

Example open-ended questions

- What did you like best about the UI?
- What did you find most difficult or challenging?
- How might the UI better support what you're trying to do?

Piloting study design

- Dress rehearsal for conducting actual study
- Goals
  - Ensure software / prototype won't "blow up"
  - Test tasks - ensure right length & difficulty
  - Test that materials are comprehensive and comprehensible
  - Test data collection protocol and methods
- As-needed piloting
  - Use first study session as pilot only if issues arise and must be addressed

Conducting the study

Introduction

- Greet participants, introduce yourself, thank them
- Build rapport, socialize
- Introduce them to the setup
- Relieve anxiety and curiosity as much as possible
- Make clear evaluating design, not participant
- Let participants know you can't answer questions about how to do task

Starting session

- Give participants description of task
- Start any video recording
- Start encouraging participant to think aloud
- Begin observing participants work on task

Interactions during the task

- Goal: listen, not talk
- Prompt participants to think aloud when necessary
  - e.g., What are you trying to do? What did you expect to happen?
- If show signs of stress / fatigue, let them take a break
- Keep participants at ease
  - If participants frustrated, reassure & calm participants
  - If so frustrated they want to quit, let them

Giving help

- If participants totally off track, small reminder of goal might help
- Should **not** give participants information about how to complete the task
- What if user asks for help?
  - Direct them to think through it or work it out for themselves

Collecting critical incidents

- *Any action that does not lead to progress in performing the desired task*
- May sometimes be related to a gulf of execution or gulf of evaluation
- Generally does not include
  - accessing help
  - random acts of curiosity or exploration
  - slips

Understanding a critical incident

- Important to understand in the moment what users goal is and what actions they are taking
- When a critical incident occurs, jot down
  - The time
  - What user was trying to do
  - What user did

Wrapping up the study session

- Provide questionnaire (if applicable) / conduct interview (if applicable)
- Answer any lingering questions the participant may have
- Thank the participant!!
- Provide any incentives (if applicable)

Reset study environment

- Make sure study environment is in the same state for all participants
  - Reset browser history / cache (if applicable)
  - Delete any user created content or materials

Analyzing data

Critical incident analysis

- Identify critical incidents where something went wrong
- Easiest to catch in the moment - important to take good notes
- Going back and looking at screencast can help you study context of issue in more detail

Reporting a critical incident

- Problem statement: summary of problem and effect on user (but not a solution!)
- User goals: what was user trying to do?
- Immediate intention: at the moment in time when problem occurred, what was the user trying to do
- Possible causes: speculate on what might have led user to take action they did

Consolidating critical incidents

- Match similar critical incidents within and across study sessions
- Identify underlying cause
- Brainstorm potential fixes

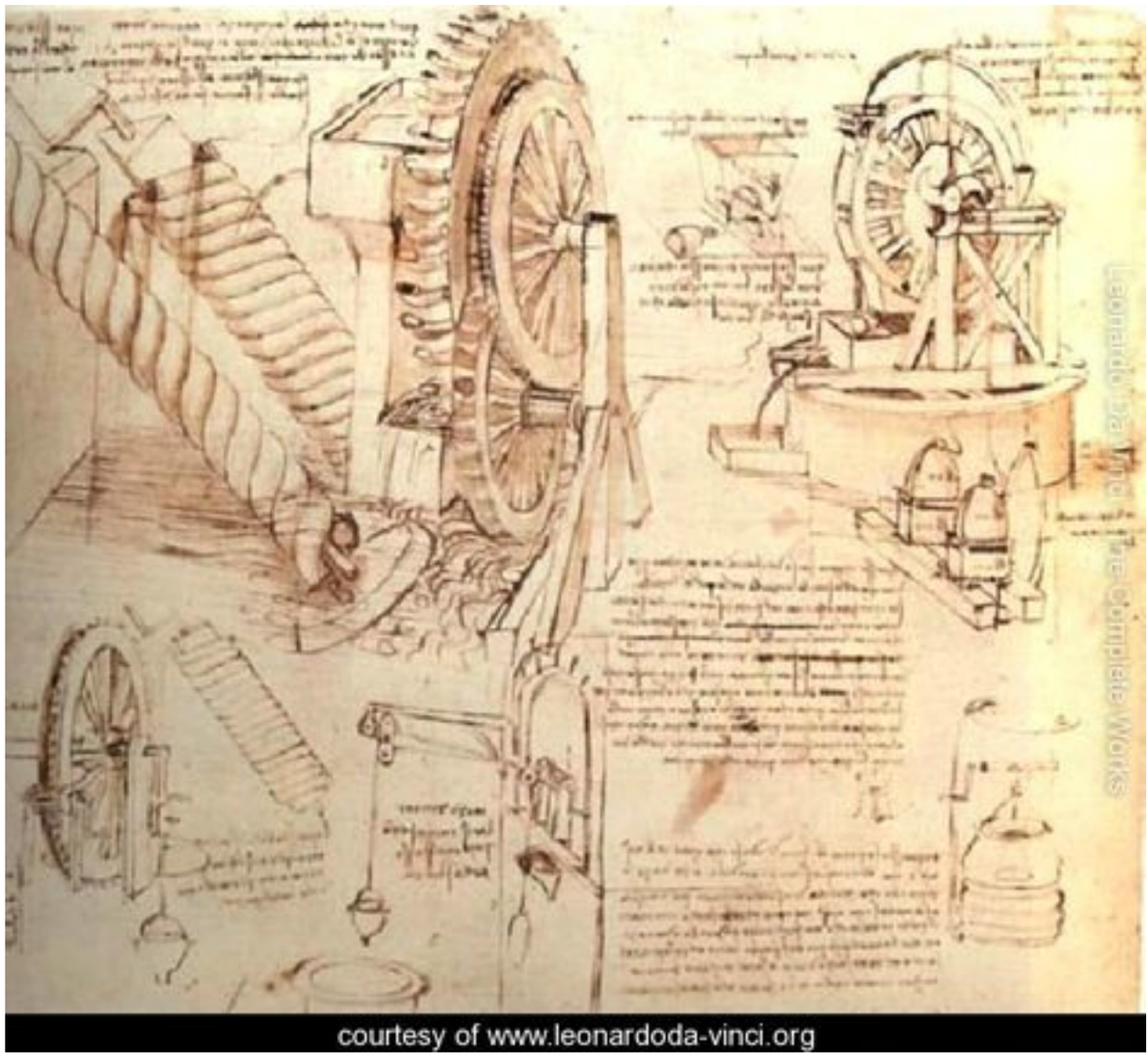
Example

				Uhh, so where did my StatusBar go?
				So my StatusBar, I'm trying to figure out who calls updateCaretStatus
		SwitchWindow	updateCaretStatus()+	Um, [rifling papers]
				Um, the callers are CaretHandler, [writing]
		ReferencesTo	View.CaretHandler.caretUpdate()+	um, caretUpdate, and that's on line 7251 ???
				and I'm going to comment that out
5:34		Edit	comments out updateCaretStatus() call	Ohh, it doesn't what me want to comment it out because it's the only thing
		Edit	comments out guards	in here, so I'm going ???? Yeah, no ???
				Next guy
				No, no what are you complaining about? [still errors in View class, but not in his metho
		ErrorTo	View.getViewConfig()+	[uses error on gutter to navigate]
				[ViewConfig can't be resolved]
				Now, I'm getting compile errors
		BackTo	updateCaretStatus()+	
		SwitchWindow	View.getViewConfig()+	**Coding bug??**
				trying to edit the thing out, trying to comment the thing out
		BackTo	View.CaretHandler.caretUpdate()+	
5:35		Edit	puts guards back, adds logging statement	Ok, he's still got an error, what's your error? Ok, he's gone
		Critique		Boy, these files are so darn long they take an incremental compiler forever to
				keep up. Another reason kids not to write files that are
				2000 lines in them, uh
				Ok, next is handleEditPaneUpdate, um
		ReferencesTo	View.handleEditPaneUpdate()+	[writing]
		Edit	comment out call to updateCaretStatus()	1671 and
		ReferencesTo	View.setEditPane()+	33
		Edit	comment out call to updateCaretStatus()	
5:36		ReferencesTo	StatusBar.handleMessage()+	Umm, this is
				StatusBar
		Edit	comment out call to updateCaretStatus()	**Breakdown?? - he's commenting out the call he just added**
		SwitchWindow	View.setEditPane()+	All right
		Run JEdit		here goes nothing
				There it is. Does it load up another file for me, no just one
				?? [creates new buffer]
		Hit breakpoint	updateCaretStatus()+	**This call is from scrolledVertically(), which didn't show up on updateCaretSt**
				Is this an Eclipse bug????
				Or is it because there were errors in the file when the query was run?
	Got a wrong answer because			Mm. I didn't get all of them?
5:37	of a tool breakdown	StackTo	View.ScrollHandler.scrolledVertically()+	What? **Very baffled about this**
				Ok, so now this is baffling me, because what I did was that I asked Eclipse to
				tell me all of the people that call updateCaretStatus(), and it gave me the list,
				and I commented out all of these, and now I'm seeing scrollHandler is calling updateCaretStatu
				and he doesn't appear on my list of people that allegedly for calling it
				Um, so, what I'm going to do go back to StatusBar, and go to updateCaretStatus
				and I'm going to call this darn thing again

Sketching & Prototyping

Why sketch?

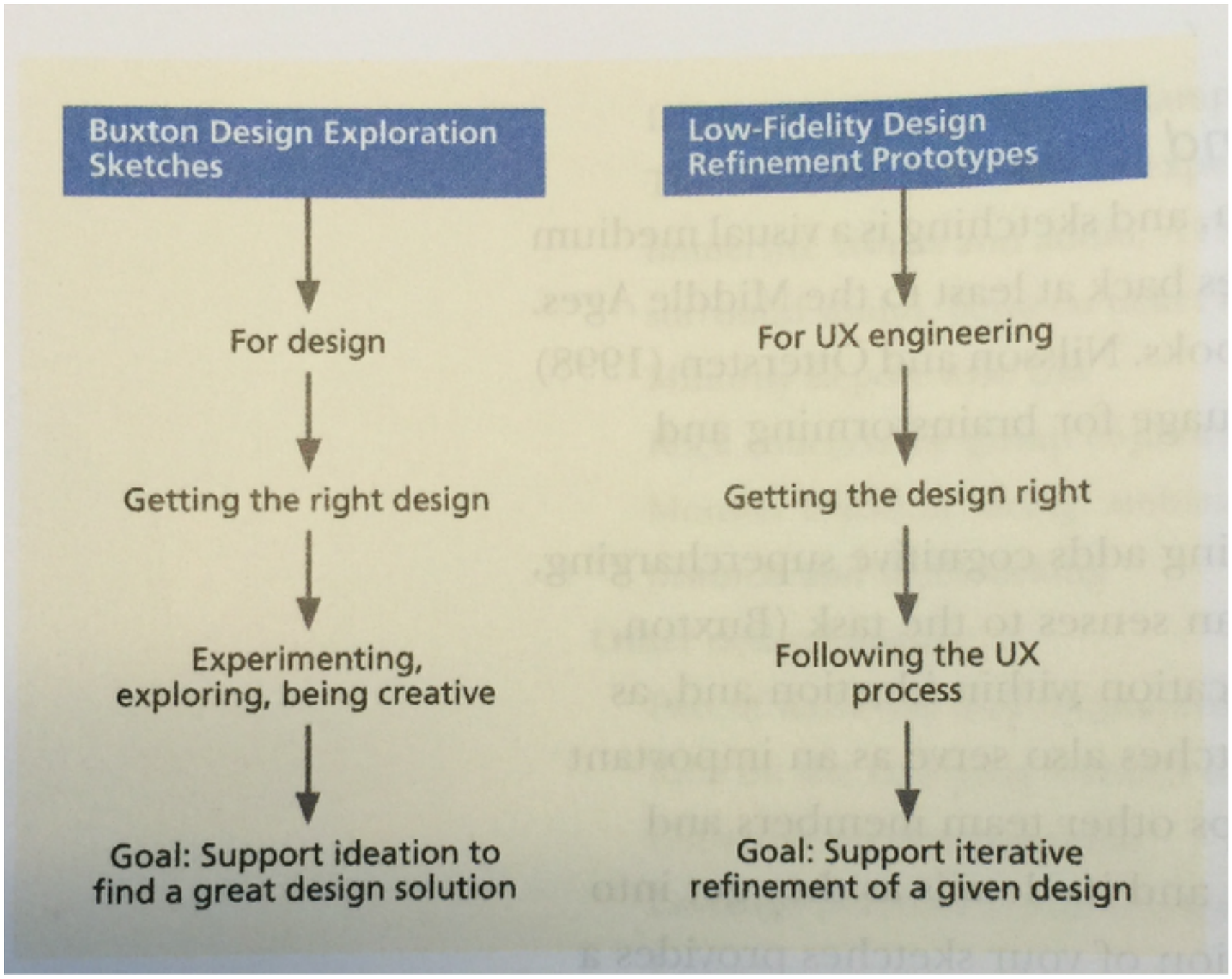
- Design is process of creation & **exploration**
- Sketching offers **visual** medium for exploration, offering cognitive scaffolding to externalize cognition
- Sketches let us explore many alternative designs



Why alternatives?

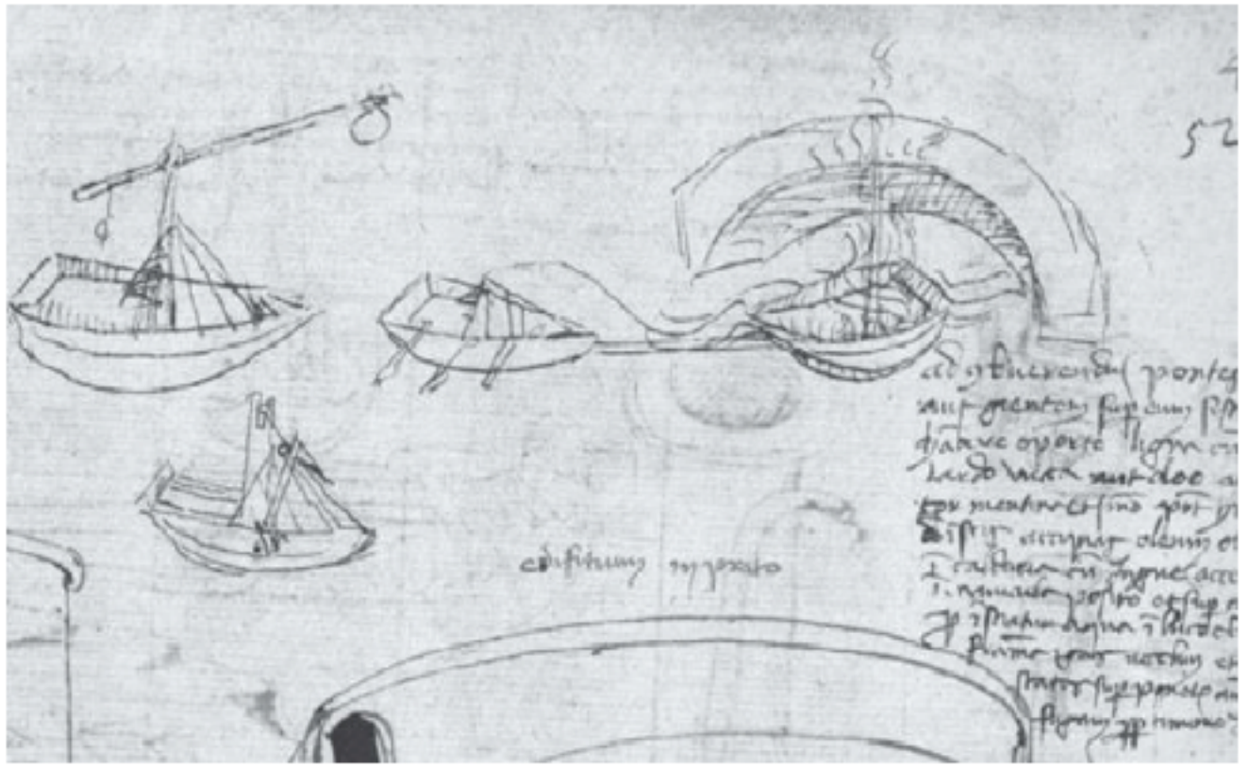
- Important to think broadly about a wide range of possible designs
  - What are the different ways in which user might do *x?*
- Rather than reimplement the status quo, alternatives offer options for doing things differently, enabling analysis of which is best
  - Important to challenge preconceptions and think deeper
- Rather than develop a single idea, sketching enables exploration and consideration of multiple designs, allowing examination of pros and cons
- Expert designers often create **many** alternatives
  - 10, 50, 100 alternative designs

Sketching vs. Prototyping



Sketches are Sketchy

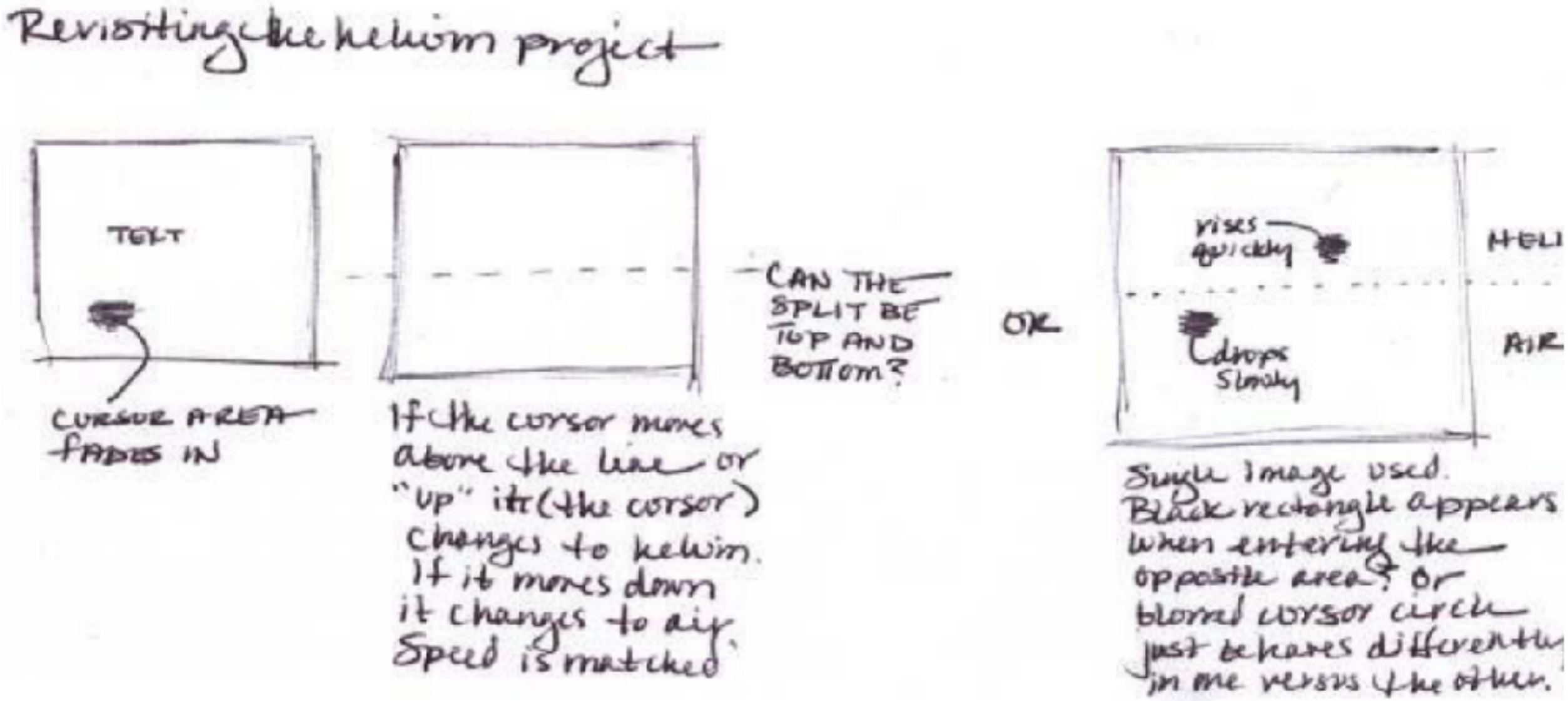
- Not mechanically correct and perfectly straight lines
- **Freehand**, open gestures
- Strokes may miss connections
- Resolution & detail **low** enough to suggest is concept
- Deliberately **ambiguous** & abstract, leaving “holes” for imagination



Benefits of Sketching

- No “programming” needed! Fast turnaround
  - Costs less
  - Allows more iterations
- Human computer
  - Can be (re)programmed quickly
  - Cannot crash
  - Changes can be made on the fly
- Developers feel less affection for status quo because changes are easy
- Rough “sketchy” appearance
  - Emphasizes content instead of appearance
  - Avoids low-level critiques of visual detail
  - Users are more willing to criticize high-level problems and less willing to blame themselves if something doesn’t work

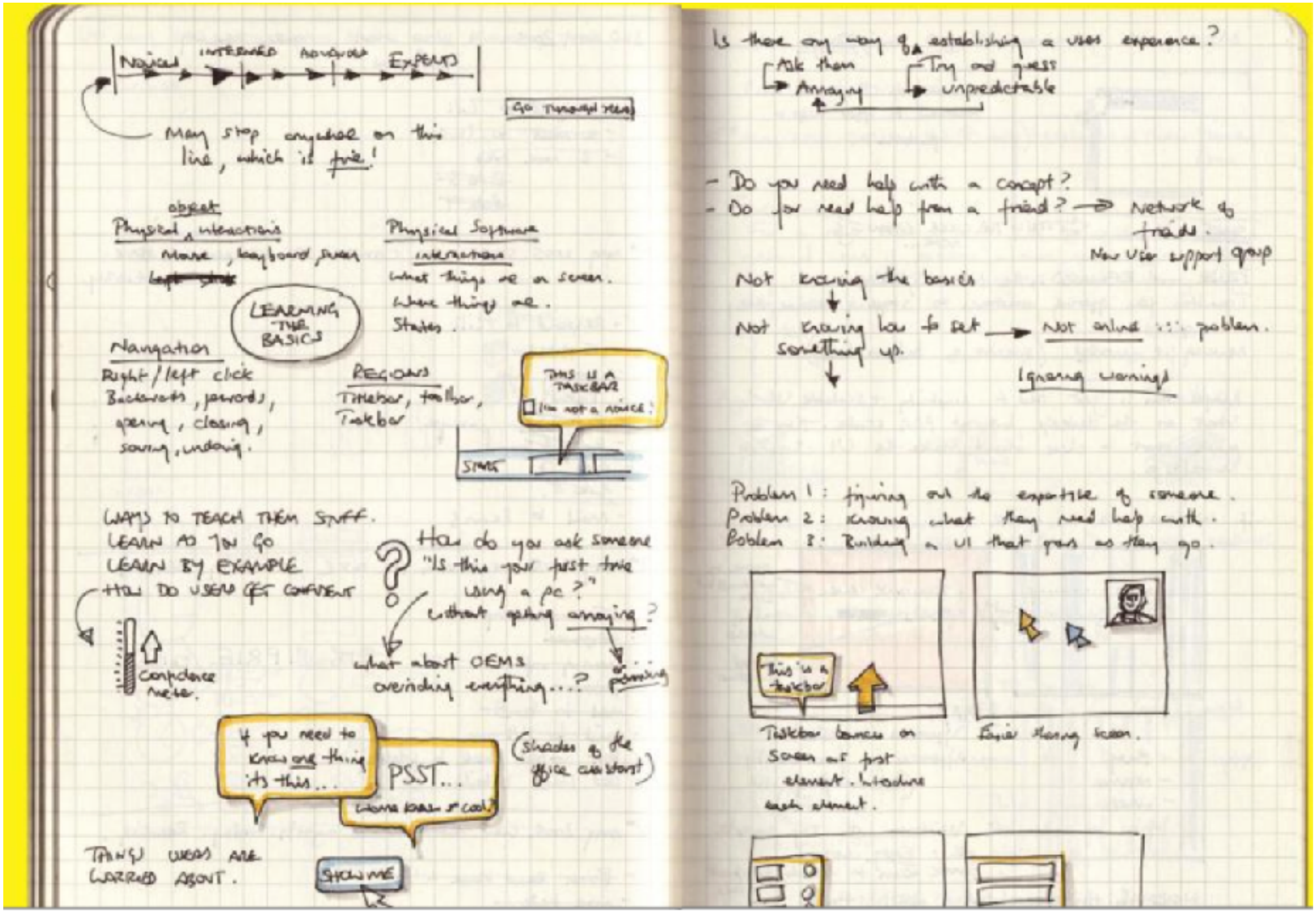
Sketches include annotations



Myers et al. (2008). How Designers Design and Program Interactive Behaviors. *VL/HCC 2008.*

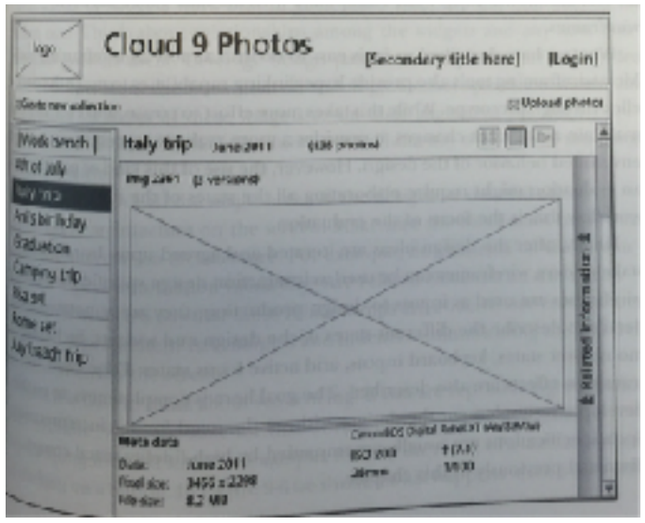
- Annotations explain what is going on in each part of sketch & how

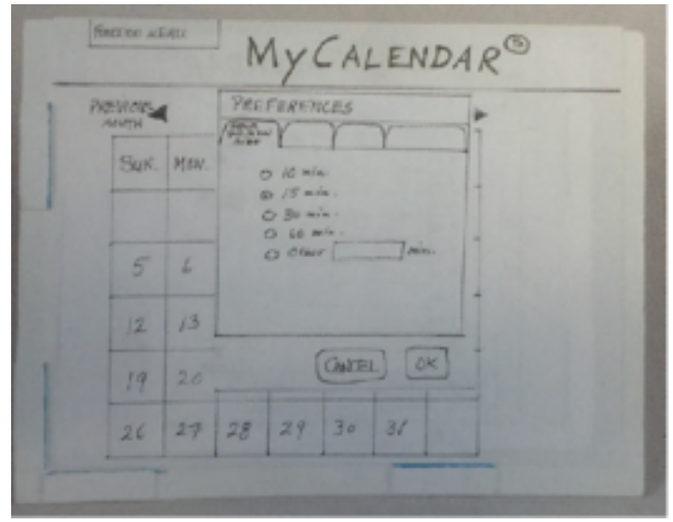
Sketches support design exploration



B. Buxton. Sketching User Experiences.

Fidelity of sketches & mockups





storyboard — wireframe — prototype

low ⟷ high

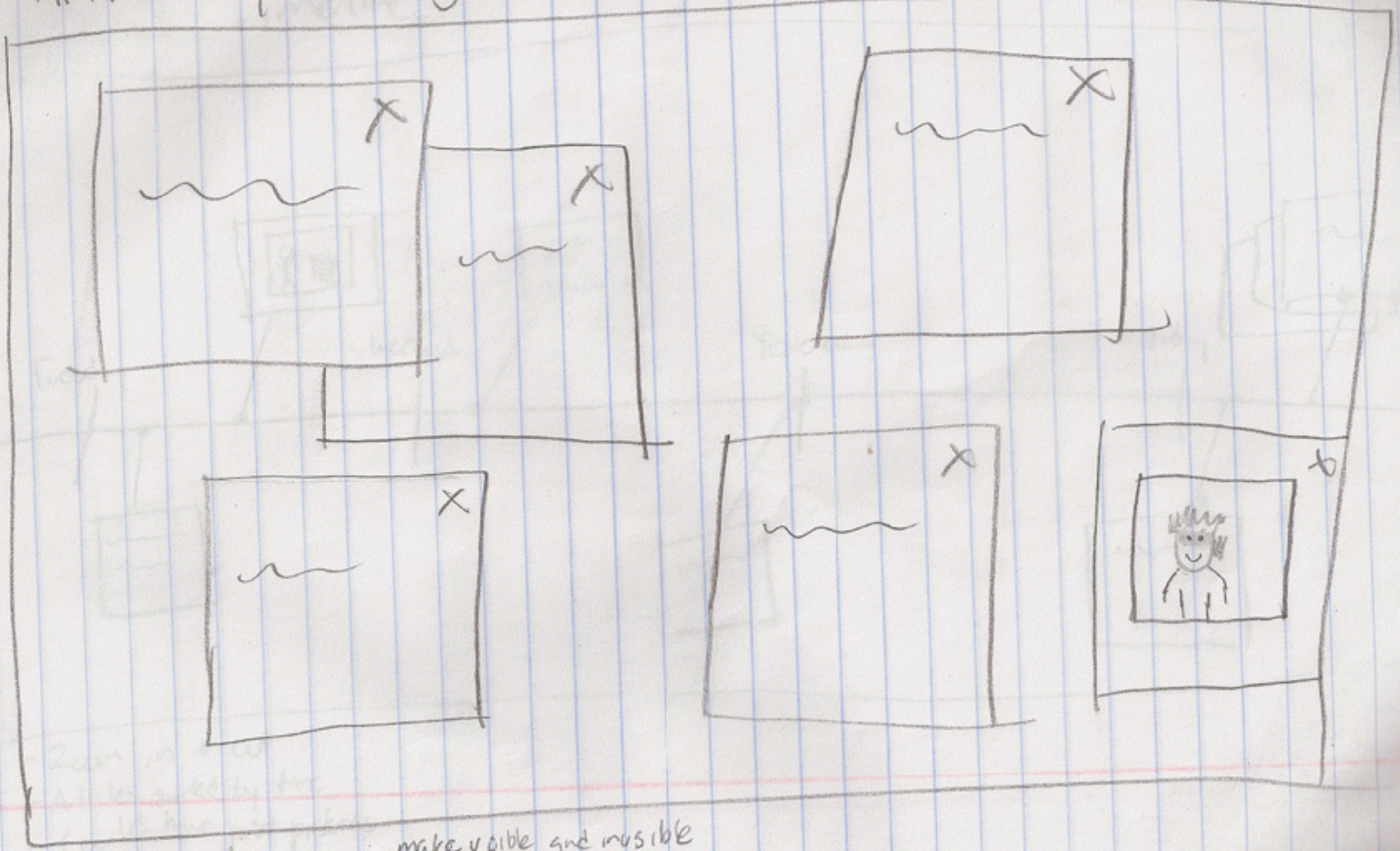
fidelity

(many details left unspecified)

(more polished & detailed)

Sketching Example: News Viewer

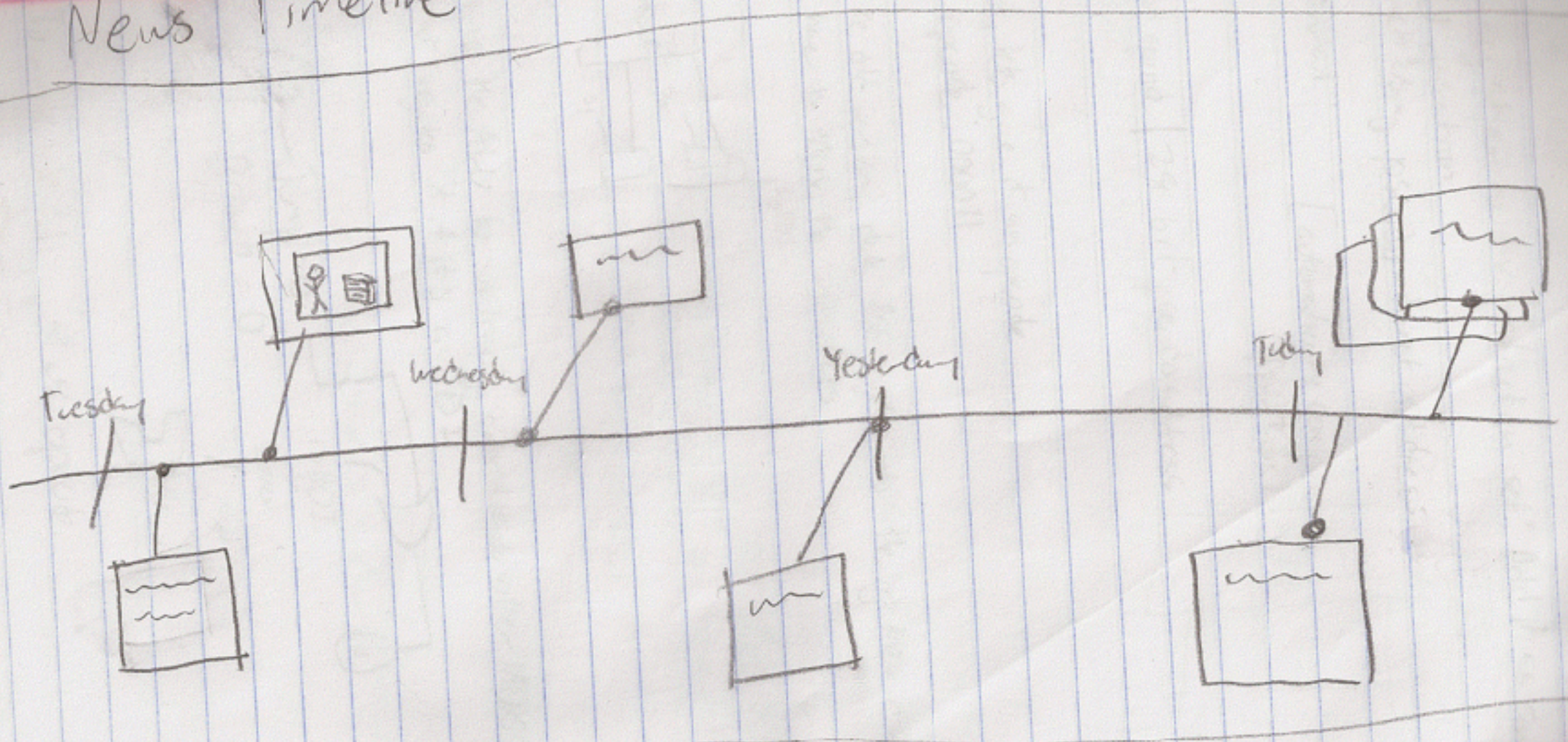
Article Layout through moveable windows (DADA) - drag and drop articles



- Moveable windows
- closeable
- Layered by importance

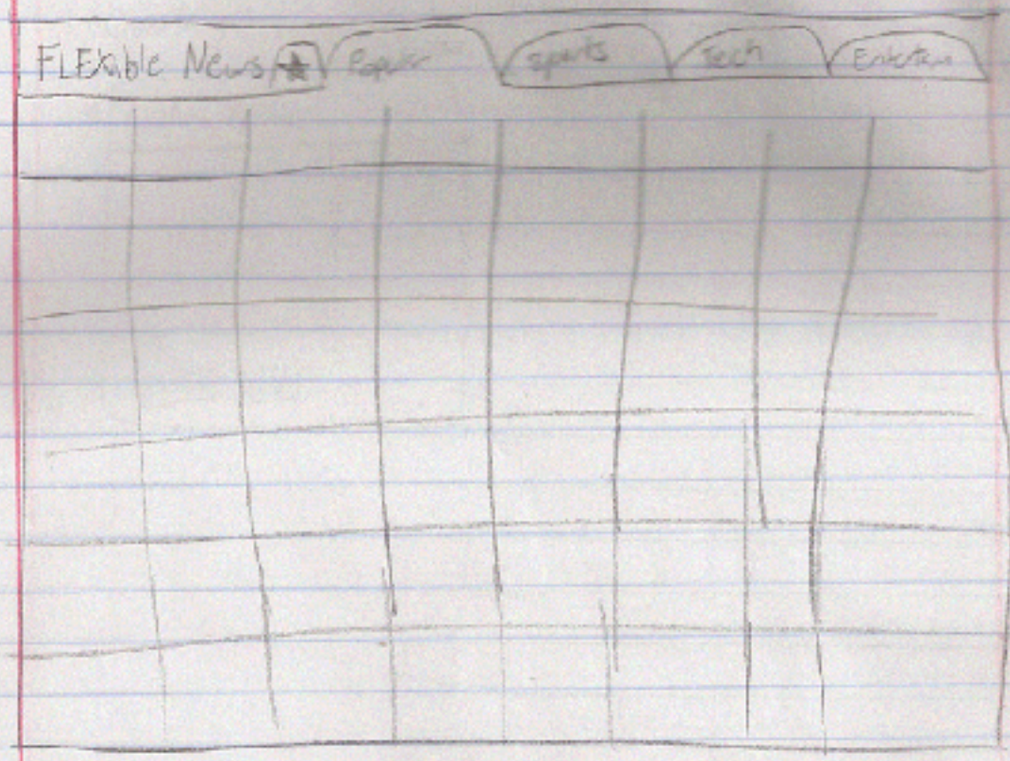
- make visible and invisible

News Timeline

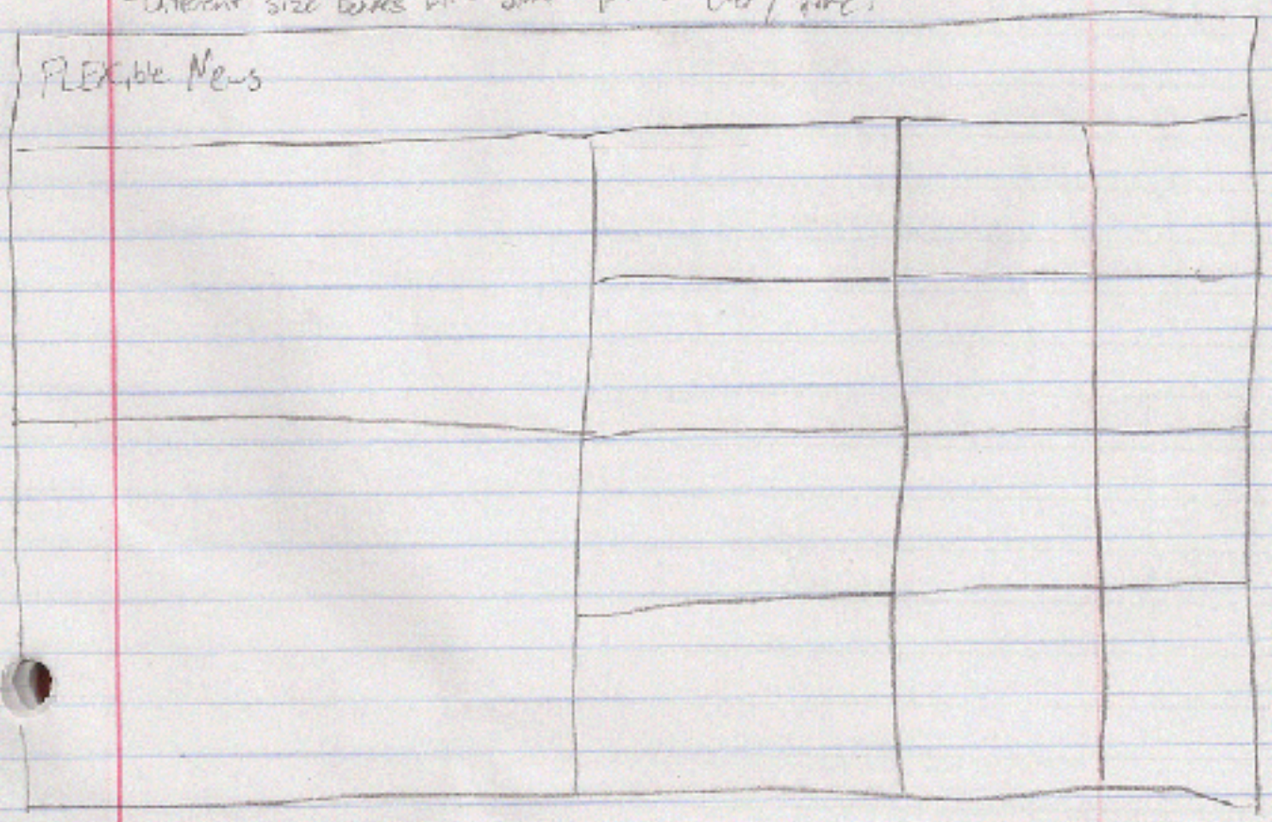


- Zoom in & out
- Articles sorted by time
- Could have just pictures

UID Wireframe



- Even boxes?
- Different size boxes with similar format every time?



Storyboards for UI design

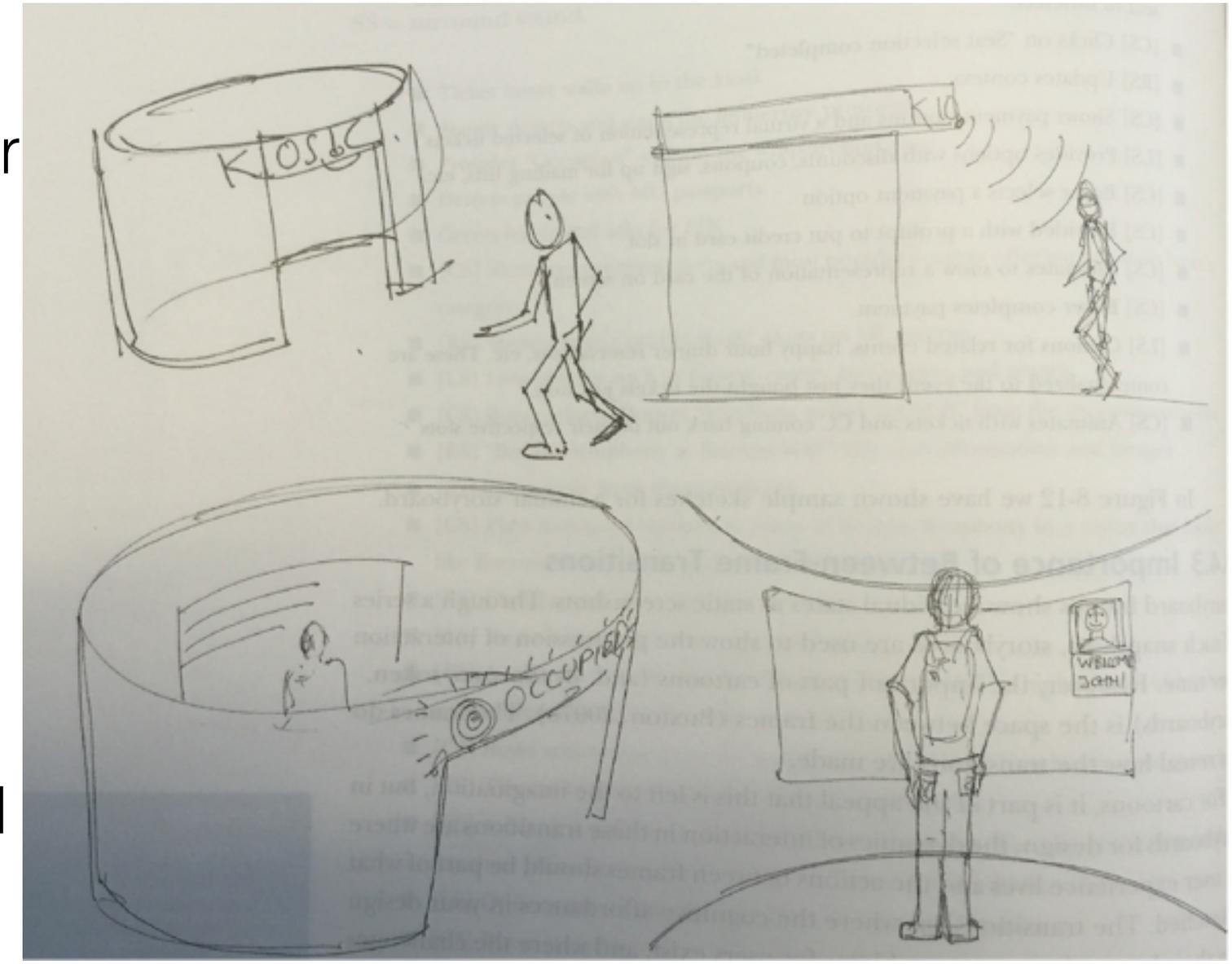
- Sequence of visual “frames” illustrating **interplay** between user & envisioned system
- Explains how app fits into a larger **context** through a single scenario / story
- Bring design to **life** in graphical clips - freeze frame sketches of user interactions
- “Comic-book” style **illustration** of a scenario, with actors, screens, interaction, & dialog

Crafting a storyboard

- Set the stage:
  - Who? What Where? Why? When?
- Show key interactions with application
- Show consequences of taking actions
- May also think about errors

Example: ticket kiosk

Ticket buyer walks up to the kiosk



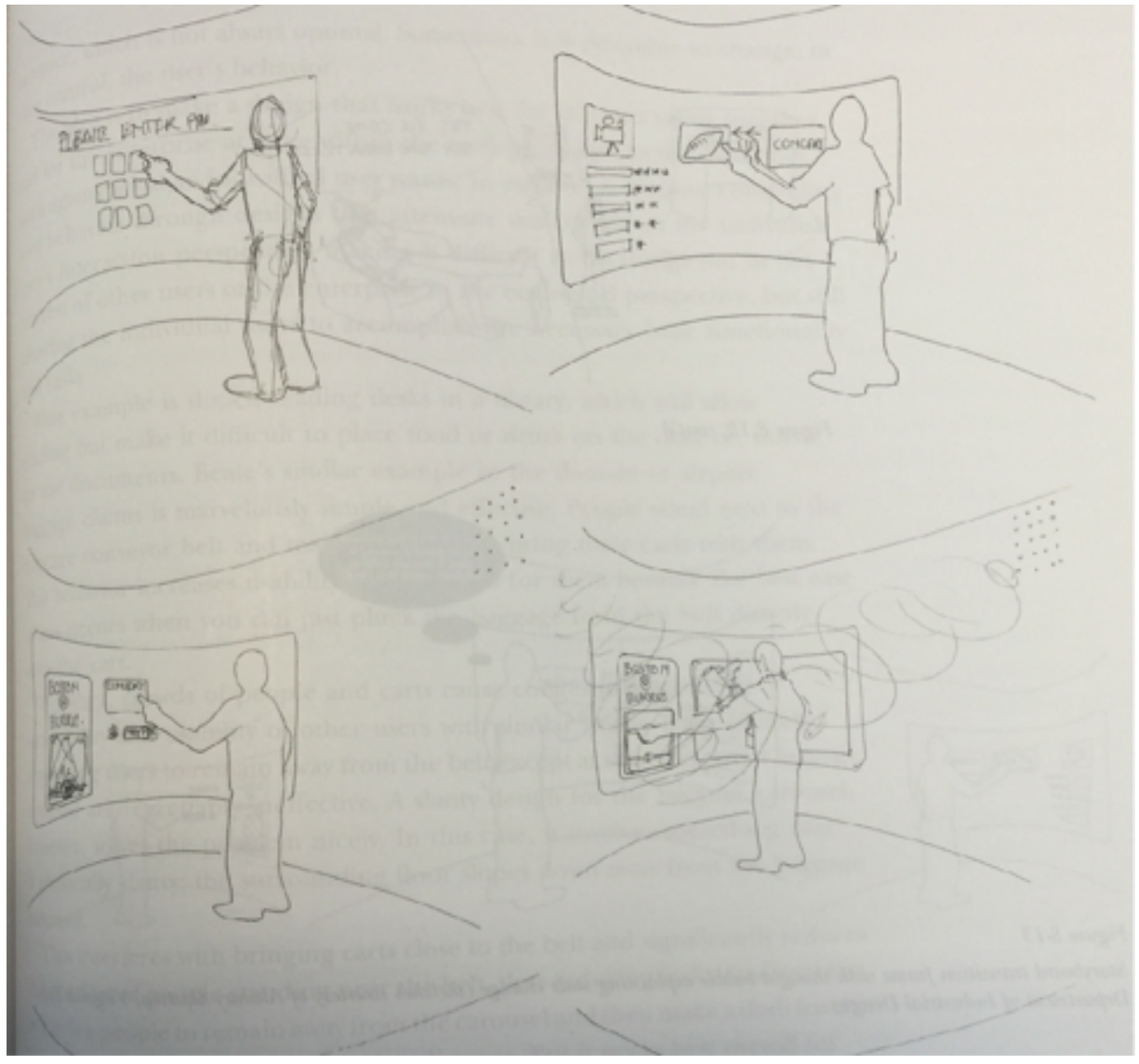
Sensor detects user & starts immersive process

Displays “Occupied” sign on wraparound case

Detects people with ID card

Example: ticket kiosk

Greets buyer and asks for PIN



Shows recommendations & most popular categories

Buyer selects “Boston symphony at Burruss Hall”

Plays music from symphony, shows date & time picker

Frame transitions

- Transitions between frames particularly important
- What users think, how users choose actions
- Many problems can occur here (e.g., gulfs of execution & evaluation)
- Useful to think about how these work, can add thought bubbles to describe

Think-Aloud Usability Study

- In groups of 2
  - Conduct a small think-aloud usability study.
  - One person will serve as participant. Other as observer.
  - Observer will ask participant to complete a short programming task.
  - Observer takes notes on activity, notes key steps and any critical incidents that occur.

In Class Activity