Design Thinking and Conceptual Design

SWE 432, Fall 2018
Design and Implementation of Software for the Web
Review: What usability is not

• Not “dummy proofing”
• Not being “user-friendly”
• Not just “usability testing”
• Not just making software pretty
Review: Heuristics

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition vs. recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation
Review: Heuristic Example

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition vs. recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation
Today

- How do we set ourselves up to build good interfaces from the start?
- What is the iterative process by which we start out with a lot of ideas, and end up with some good, end result interface?
- Case study in design - Piles
- Reminder: HW4 due 11/14

For further reading on the case study: https://dl.acm.org/citation.cfm?id=143055
Different designs for the same problem
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

**Buxton Design Exploration Sketches**
- For design
- Getting the right design
- Experimenting, exploring, being creative
- Goal: Support ideation to find a great design solution

**Low-Fidelity Design Refinement Prototypes**
- For UX engineering
- Getting the design right
- Following the UX process
- Goal: Support iterative refinement of a given design
Physical sketches

• Production tools for sketching:
  • whiteboards, blackboards, cork boards, flip chart easels
  • post it notes
  • duct tape, scotch tape, push pins, staples
  • marking pens, crayons, spray paint
  • scissors, hobby knives, foam core board
  • duct tape
  • bits of cloth, rubber
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
Sketch Example

Sketch: List

Sketch: Icons
Rules for sketching

• Everyone can sketch; you do not have to be artistic
• Most ideas conveyed more effectively with sketch than words.
• Sketches are quick and inexpensive to create; do not inhibit early exploration
• Sketches are disposable; no investment in sketch itself
• Sketches are timely; made in-the-moment, just-in-time
• Sketches are plentiful; entertain large # of ideas w/multiple sketches of each
Sketches include annotations


• Annotations explain what is going on in each part of sketch & how
Sketches support design exploration

Is there any way of establishing a users experience?
Ask them TRY and guess Anyway unpredictable

- Do you need help with a concept?
- Do you need help from a friend?
  Network of friends
  New User Support Group

Not knowing the basics
Not knowing how to set something up
  Not online... problem
  Ignoring warning

Problem 1: figuring out the expertise of someone.
Problem 2: knowing what they need help with.
Problem 3: building a UI that goes as they go.

LEARNING THE BASICS

Physical software
- Interaction
  Mouse, keyboard, etc.
  What things are on screen.
  Where things are.
  States.

Dave is a Teacher
I'm not a source

LEARNING THE BASICS

Regions
- Titlebar, footer, toolbar

Right / Left Click
Backwards, forwards
Opening, closing, saving, undoing.

Search

If you need to know one thing it's this...
PSST...

Things we care about

B. Buxton. Sketching User Experiences.
Fidelity of sketches & mockups

- **Storyboard**: low (many details left unspecified)
- **Wireframe**: (more polished & detailed)
- **Prototype**: high
Sketching Example: News Viewer
Article Layout through movable windows (DADA) - drag and drop articles

- Movable windows
- Closeable
- Layered by importance

- Make visible and invisible
News Timeline

- 2am: in & cut
- Articles sorted by time
- Could have just pictures
**UID Wireframe**

- Flexible News
- Tech
- Education

<table>
<thead>
<tr>
<th>Even boxes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Different size boxes with single theme every time?</td>
</tr>
</tbody>
</table>

**Flexible News**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Article Layout through multiple windows (DADA): drag and drop articles

- Multiple windows
- Transparency
- Import by importance

News Timeline

- Run in or out
- Articles sorted by size
- Child the post photos

UID Wireframe

- Flexible News
- Paper
- Sports
- Tech
- Finance
- Events

- Even boxes?
- Different size boxes with similar format every time?
Conceptual design
Designer’s mental model

• Conceptualization of the envisioned system
  • what it is
  • how it is organized
  • what it does
  • how it works
User’s mental model

- Comes up from existing interactions with systems
- Users form cause & effect relationships to form theories that guide actions
Conceptual design

- **Goal:** match users’ **mental model**
- **Tool:** Metaphor - analogies from existing system
  - Offers expectations about what system does & what can be done
- **Examples**
  - Email ←→ physical mail
  - Backup software ←→ time machine
  - OS desktop ←→ top of a desk
Storyboards
Storyboard for Disney’s Melody: Adventures in Music (1953)
Source: Michael Sporn Animation
1. Now STUDENTS
   LISTEN CLOSE
   TO ME -

2. CAUSE TODAY
   WE'RE GONNA TALK
   ABOUT MEL & DDE

3. NOW ALL THE
   PROFESSORS AND
   CATS HAVE FOUND
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:
- Show key interactions with application
- Show consequences of taking actions
- May also think about errors
Example elements of a UI storyboard

- Hand-sketched pictures annotated with a few words
- Sketch of user activity before or after interacting with system
- Sketches of devices & screens
- Connections with system (e.g., database connection)
- Physical user actions
- Cognitive user action in “thought balloons”
Example: ticket kiosk

Ticket buyer walks up to the kiosk

Displays “Occupied” sign on wraparound case

Sensor detects user & starts immersive process

Detects people with ID card
Example: ticket kiosk

Greets buyer and asks for PIN

Buyer selects “Boston symphony at Burruss Hall”

Shows recommendations & most popular categories

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
Design Fail

1: Auto-login to Piazza app
2: Network error
3: Asked for password
Where do we start?
Needfinding (a.k.a. design research)

- Goal: understand user’s needs
- Use of methods to gather qualitative data
  - behaviors, attitudes, aptitudes of potential and existing users
  - technical, business, and environmental contexts - domain
  - vocabulary and social aspects of domain
  - how existing products used
- Empowers team w/ credibility and authority, helping inform decisions
Interviews

• May include both current users and potential users w/ related needs

• Questions
  • context of how product fits into lives or work
  • when, why, how is or will product be used
  • what do users need to know to do jobs?
  • current tasks and activities, including those not currently supported
  • goals and motivations of using product
  • problems and frustrations with current products or systems
Observations

• Most incapable of accurately assessing own behaviors
• May avoid talking about problems to avoid feeling dumb
• Observing yields more accurate data
• Capture behaviors: notes, pictures, video (if possible)
Contextual design

• An approach to answering the question, “What should we build to help people do their work better?”

• Key elements
  • Contextual Inquiry – Gather data from customers while they do their work to help decide what system should do.
  • Work modeling – Use data to build models of work that are explicit and sharable.
  • Work redesign – Use data and models to design work model for how customers will work in the future. The core design problem is work design, rather than technology design.
  • Use work model and iterative prototyping to drive detailed user interface design.
  • Customer data, modeling and work design drive technology design.
Design Thinking Case Study - Piles

- The context: Apple Computer, 1992, 3 researchers (Richard Mander, Gitta Salomon and Yin Yin Wang)
- The design problem: How should computers help users organize and file information?
- The method: How do users organize and file information best without computers?

Computer users are confronted with large amounts of information, but currently are only provided with a hierarchical filing system for managing it [folders].
Piles - Methodology

• Interview 13 Apple employees in different departments for 30-60 minutes
• Studied:
  • How does information arrive in your work area?
  • What do you do with information once you get it?
  • Where does it go next?
  • Where and how is it stored?
  • How do you work with (or would work with) an assistant?
Piles - Study Results

• Users used many artifacts like:
  • Folders
  • File cabinets
  • File racks
  • Binders
  • Piles

• Problems with filing in folders/cabinets:
  • “I’m not always as good at categorizing things as I would like...it’s hard to get it right and I’m sort of a perfectionist, so I think that I should know exactly how I should do it...I like things in their place, but I can’t figure out exactly what place.”
“Seemingly disordered piles were often sensible to the person who created them, because they developed through many interactions over a long period of time.”

“…Most workers kept information they needed in a specific working area. A common strategy was to create separate piles for each project and place them within the working area, at distances that reflected their urgency.”
Piles as a Design Metaphor

- Insight: Many tasks do not rely on hierarchy of files (e.g. organization in folders in cabinets)
  - Note, not first to suggest metaphor of piles, Thomas Malone (Xerox PARC) described nearly 10 years earlier in 1983

- How do workers use piles?
  - Edge browse - find cues from the edge of a pile (thickness, color, texture)
  - Restack - Start at the top, browse down by removing things
  - Hinge - Unfold the pile like a binder
  - Spread out - See all/many items at once
Assistance with Information Management

• Most participants in the study did not have an assistant but mentioned that they wanted one, why?

• Assistants might:
  • Sort incoming data into categories
  • Filtered incoming data
  • Create piles
  • All **in collaboration with** the worker
    • “I’ll go into his office and put [labels] on piles on his floor and he’ll look at it and say ‘no’ or he’ll say ‘that’s pretty good’.”
Piles - Sketches

• Created sketches to facilitate discussion and evaluation

• Example features:
  • System-created piles

User-created pile (messy)  System-created pile (Organized)
Piles - Sketches

- Created sketches to facilitate discussion and evaluation
- Example features:
  - New files added to the pile are directly visually represented
Piles - Sketches

• Created sketches to facilitate discussion and evaluation
• Example features:
  • System-user collaboration for defining rules

Select an item from a pile  Move to new pile  Update pile script
Piles - Sketches

• Created sketches to facilitate discussion and evaluation

• Example features:
  • Browsing by spreading a pile out
Piles - Sketches

- Created sketches to facilitate discussion and evaluation
- Example features:
  - Browsing and maintaining structure (kind of like hinge)
Piles - Sketches

• Created sketches to facilitate discussion and evaluation
• Example features:
  • Visualizing the contents of a pile

All data, ordered and colored by date
All data, ordered and colored by date, piled by content
Piles - Evaluation

• Built prototypes, conducted studies
• We’ll return to this when we discuss prototyping and user studies
• What do YOU think are good/bad things about this metaphor?
Piles - Legacy

- Patent issued to Apple in 2001
- 2007 (OS 10.5) introduced Cover Flow
Piles - Legacy

- Patent issued to Apple in 2001
- 2007 (OS 10.5) also introduced stacks
Piles - Legacy

- Patent issued to Apple in 2001
- 2018 (OS 10.14) introduces desktop stacks
Desktop Stacks

Stack by kind
Stack by last opened
Stack by tag

Open a stack