Sketching and Prototyping

SWE 432, Fall 2019
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

• More on heuristic evaluation
• 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?

For further reading on the case study:
https://dl.acm.org/citation.cfm?id=143055
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
In class activity

- Form groups of 2
- Together select an application or website (e.g., Word, Twitter)
- Work individually identify at least 3 usability issues
- For each issue, identify the heuristic, identify the functionality in the application, and summarize how the heuristic is violated.
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
Using heuristic evaluation

- Can be used informally to identify issues in a website
- Can be used as a more formal usability inspection method
- Evaluators each first separately identify issues
- Issues then combined from each evaluator
Heuristic evaluation in groups

Figure 2: Average proportion of usability problems found as a function of number of evaluators in a group performing the heuristic evaluation.
Advantages of HE

- “Discount usability engineering” - Intimidation low
- Don’t need to identify tasks, activities
- Can identify some fairly obvious fixes
- Can expose problems user testing doesn’t expose
- Provides a language for justifying usability recommendations
Disadvantages of HE

- Un-validated
- Do not employ real users
- Can be error prone
- Better to use usability experts
- Problems unconnected with tasks
- Heuristics may be hard to apply to new technology
Ways to use HE

• Early in design process to catch major issues
• When time or resources are not available for empirical usability evaluation
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 user 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

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

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

Things users are learned about

GMU SWE 432 Fall 2019
B. Buxton. Sketching User Experiences.
Fidelity of sketches & mockups

**storyboard**

- low
- (many details left unspecified)

**wireframe**

- fidelity

**prototype**

- high
- (more polished & detailed)
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

- Tuesday
- Wednesday
- Yesterday
- Today

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

<table>
<thead>
<tr>
<th>Flexible News</th>
<th>Paper</th>
<th>Sports</th>
<th>Tech</th>
<th>Educat</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Even boxes?
- Different size boxes with same frame every time?

Flexible News

<table>
<thead>
<tr>
<th>Flexible News</th>
<th>Paper</th>
<th>Sports</th>
<th>Tech</th>
<th>Educat</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 O'Dell.
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

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
Design Fail

1: Auto-login to Piazza app

2: Network error

3: Asked for password
Prototyping
Prototyping

• How do you know your system design is right before you invest the time to build it?

• Answer: prototyping!
  • Evaluation performed before investing resources in building finished product
  • Early version of system constructed much faster & with less expense used to evaluate & refine design ideas
# Fidelity of prototypes

<table>
<thead>
<tr>
<th>Kind of Iteration</th>
<th>Purpose</th>
<th>Types of Prototypes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideation and sketching</td>
<td>To support exploring ideas, brainstorming, and discussion (so design details are inappropriate)</td>
<td>Sketches, fast and disposable mockups, ultralow fidelity</td>
</tr>
<tr>
<td>Conceptual design</td>
<td>To support exploration and creation of conceptual design, the high-level system structure, and the overall interaction metaphor</td>
<td>Evolution from hand-drawn paper, computer-printed paper, low-fidelity wireframes, high-fidelity wireframes, to pixel-perfect interactive mockups (to communicate with customer)</td>
</tr>
<tr>
<td>Intermediate design</td>
<td>To support interaction design for tasks and task threads</td>
<td>Evolution from paper to wireframes</td>
</tr>
<tr>
<td>Detailed design</td>
<td>Support for deciding navigation details, screen design and layout, including pixel-perfect visual comps complete specification for look and feel of the “skin”</td>
<td>Detailed wireframes and/or pixel-perfect interactive mockups</td>
</tr>
<tr>
<td>Design refinement</td>
<td>To support evaluation to refine a chosen design by finding and removing as many UX problems as possible</td>
<td>Medium to high fidelity, lots of design detail, possibly a programmed prototype</td>
</tr>
</tbody>
</table>
Interactivity of prototypes

- Scripted, click through prototypes
  - Prototype w/ **clickable** links to move between screens
- Live action storyboard of screens
- Simulates real **task flow**, but w/ static content
- Fully-implemented prototypes
  - Usually **expensive** to implement actual system
  - But can build key piece of system first to evaluate
Wizard of Oz

- Goal: **simulate** actual system w/ out building it
  - Want user to interact **as if** they were interacting w/ real system
  - Helps explore how users would interact w/ novel interaction if it were to exist
- Example: natural command line (Good et al 1984)
  - Users typed in commands to interact w/ computer
  - Commands intercepted by hidden human who interpreted commands & executed them
Paper prototypes

• **Low fidelity** prototype w/ paper mockups
• Goal: get feedback from users early w/ very low cost interactive prototype of envisioned interaction design
Paper prototyping (1)

- Set a realistic deadline
- Gather set of paper prototyping materials
- Work **fast** & do not color within the lines
- Reuse existing sketches & mockups
- Make underlying paper mockups of key screens