CIS*3750 - System Analysis and Design in Applications
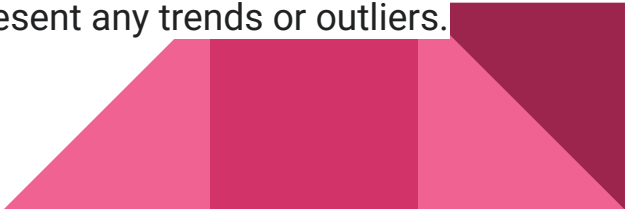
Luiza Antonie, Fall 2025, University of Guelph
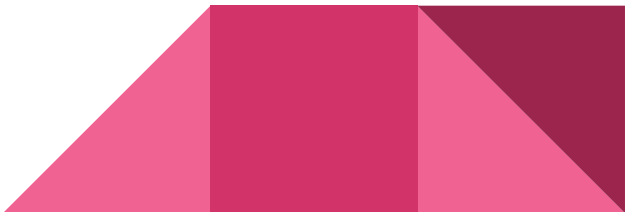
Upcoming Deadlines

- Post mortem
  - Due Oct 10
- Quiz 1
  - Oct 16, in class

Post Mortem

- Due <u>October 10</u> (submit PDF through Dropbox)
- The document should include the following:
  - Use Cases:
    - Your post mortem should include a list of the <u>Use Cases you presented</u> during each prototyping interaction. You should also indicate <u>who took part in the session</u>, and <u>what each team member did</u>. Indicate information you learned about the person taking part in the session. You could include information such as age, computer skill level, and whether they are a representative of a user or not. [2 pages maximum]
  - Stop/Start/Continue
    - Your document must include a <u>summary of the findings</u> from the paper prototyping participants, identifying the common <u>things that they thought didn't work</u> or should be changed (e.g., things to stop doing), <u>things they thought you should add</u> to the design (e.g., things to start doing), and the <u>things they thought worked</u> with your design (e.g., things to continue doing). Summarize the results and present any trends or outliers. [3 pages maximum]

Quiz 1

- Oct 16, in class
- All lectures
- All labs
- Assigned readings (see course announcements)
- Format
  - T/F
  - Multiple choice
  - Short answer

Reminders - Next Week

- No class on Tuesday (Fall break)
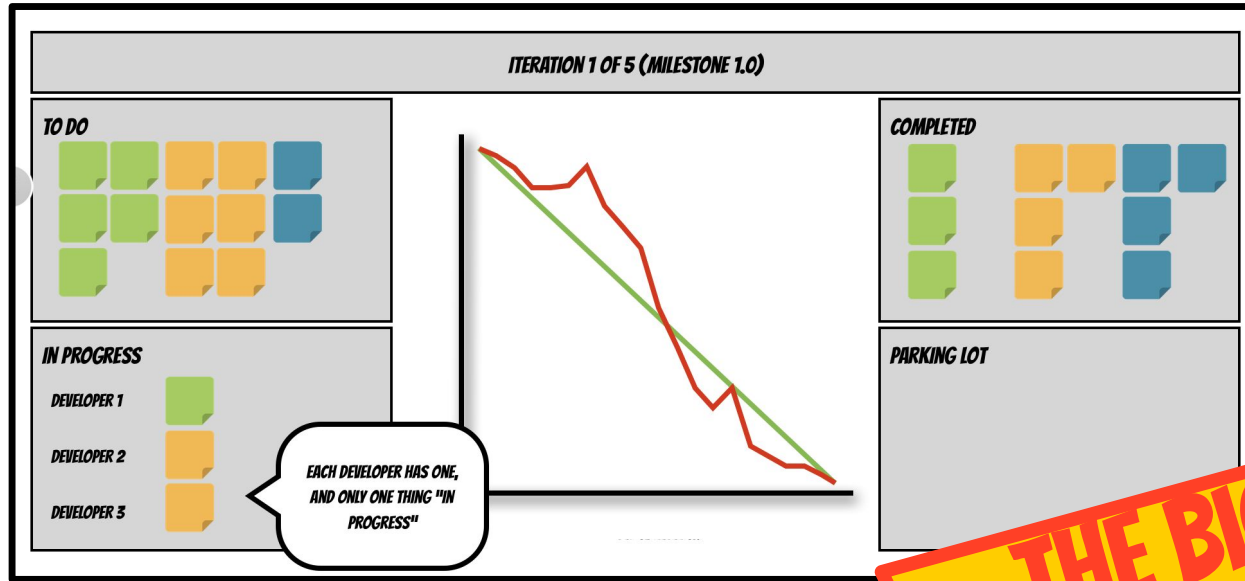- No labs next week
- Quiz 1 on Oct 16

Tasks - Planning

- Assign all the tasks in the user story to developers - add initials to sticky notes
- When a developer works on a task (and only then) - move to "in progress"
- Avoid multi-tasking unless the tasks are short and related.
- Try to finish one User Story before starting another

Planning - Burndown Chart

Sample Burndown Chart
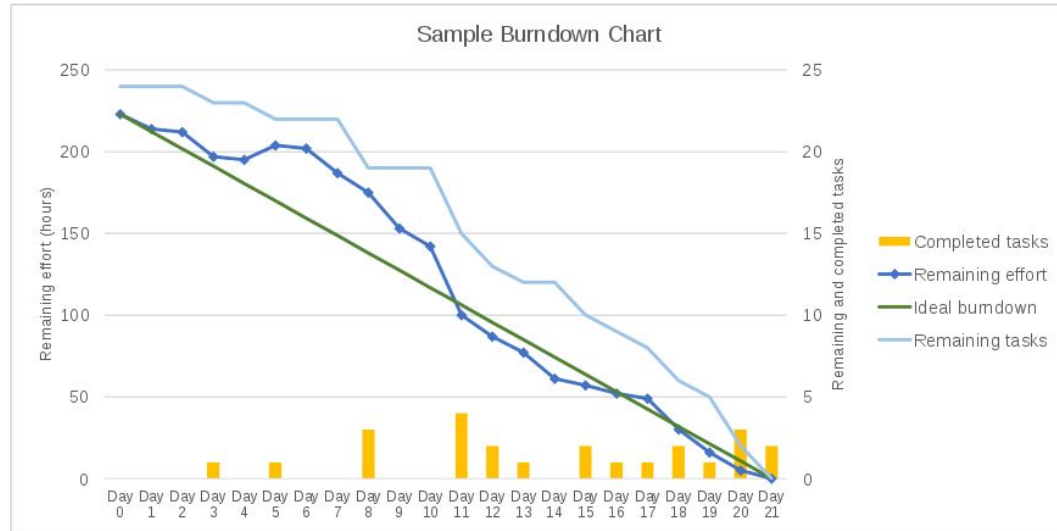
Planning

Stand-Up Meetings

- Gather around dashboard (or Trello)
- Update progress
- Update burn-down rate
- Update task status
- Review previous day's work
- Issues?
- 5-10 minutes total.

Prototyping

- What is a prototype?

- A prototype is a non-functional mock up; a model; "reasonably close".
- It does not represent necessarily ALL aspects of the program.

Why do we prototype?

- Try out design options
- Demonstrate concepts
- Usability studies

Why do we prototype?

- Validate requirements
- Explore solutions
- Validate that the UI is usable
- Help identify if proposed structure fits the user's work model

Prototypes as Communication Tools

● Developed for the user using vocabulary they understand

● Help the design team understand what the user it thinking when they use a particular word

Prototypes as Communication Tools

- Inspire "ownership", as users feel like they are co-designers in the process

- Allow iterative refinement of the process

Benefits

- Improves system usability
- Users' real needs are considered in the design process
- Improves design quality
- Reduces development effort

Types of prototypes

- Low fidelity (lo-fi)
  - paper prototyping

- High fidelity (hi-fi)
  - wireframing

Paper Prototyping

- Some things you can't learn through a paper prototyping session
  - colour, fonts
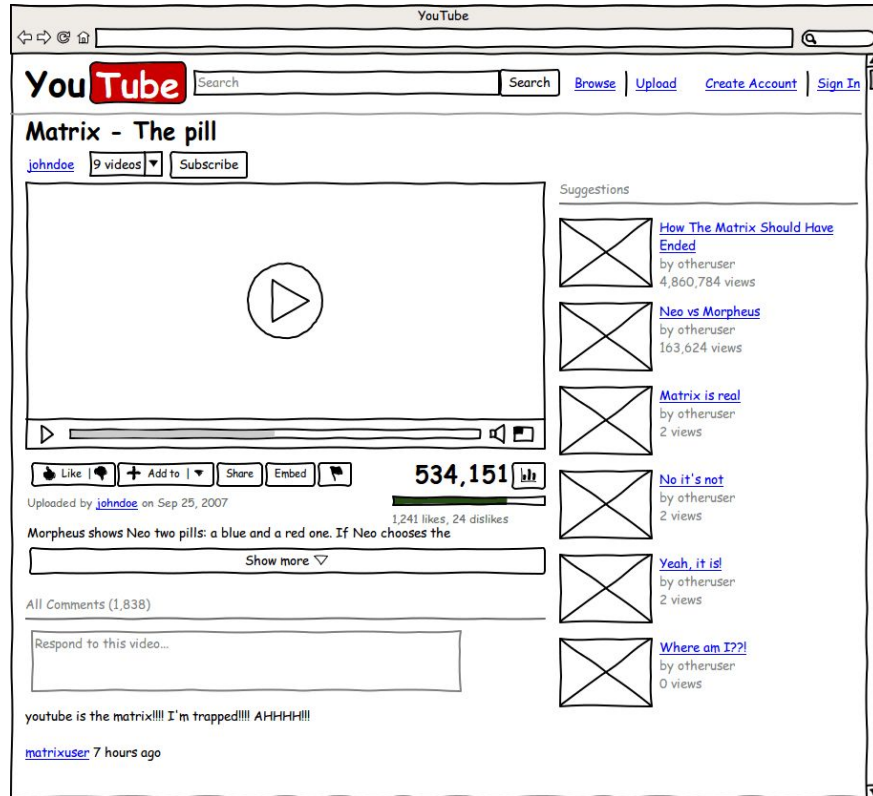 - efficiency and response time

Wireframe

- A high fidelity prototype that looks closer to the final product, and allows the user to focus on the aesthetics (and less on the process flow)
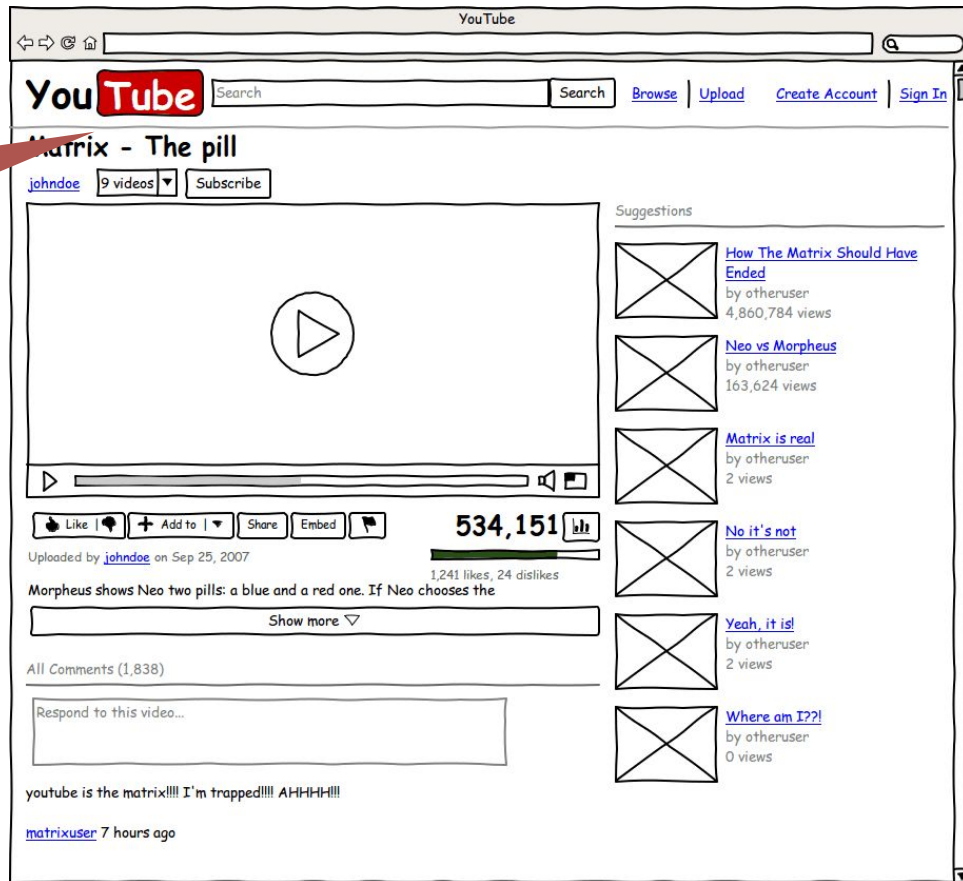
Wireframe

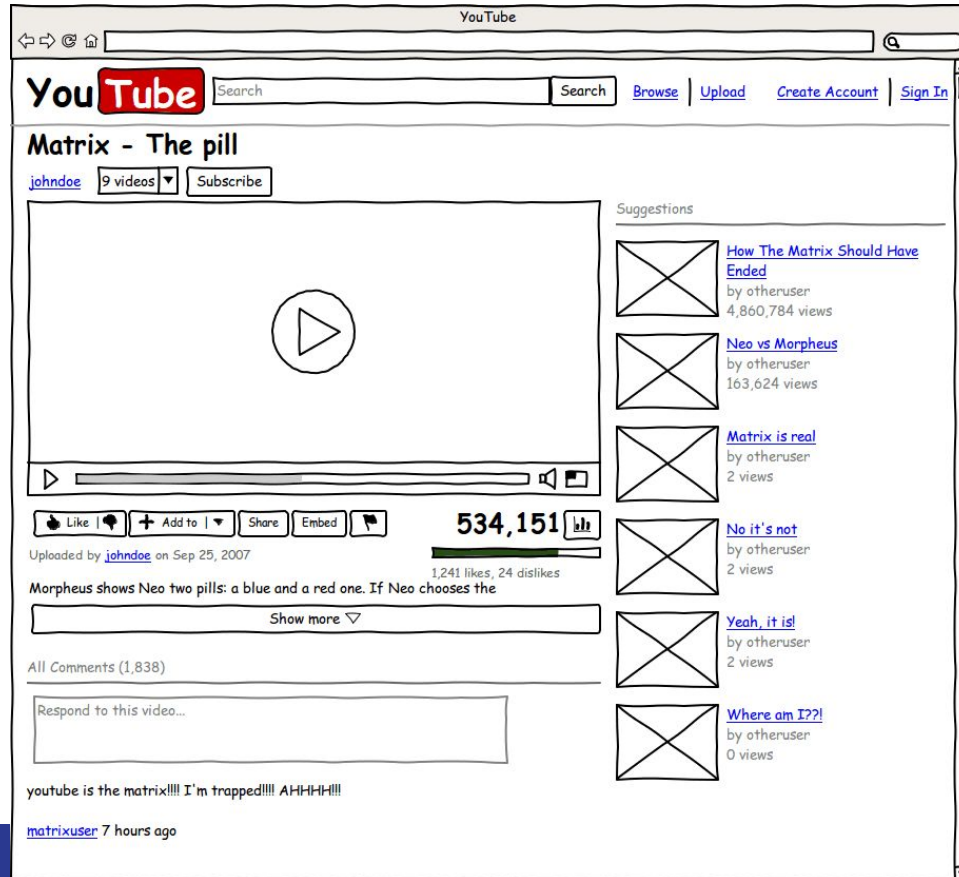

Images are typically replaces with crossed out frames

Wireframe

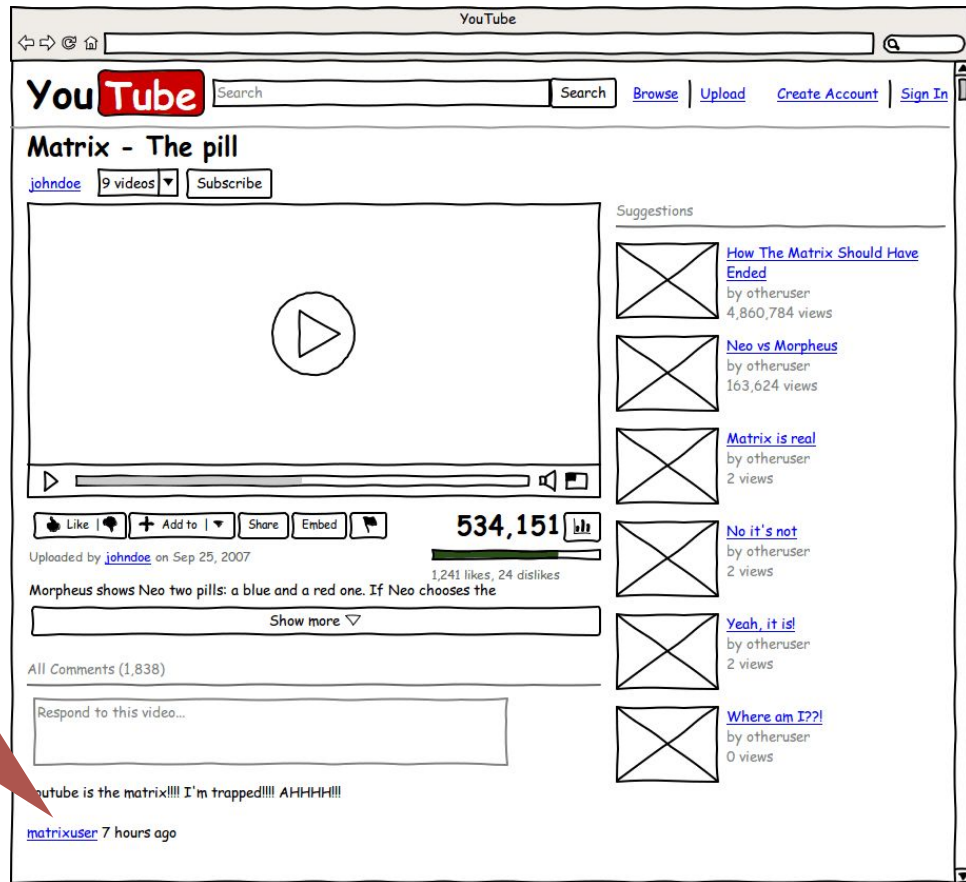Logos and some colours can be included

Wireframe

Placement of images and text are closer to the final version

Wireframe

Navigation and links are the only functionality

Wireframe

- Your wireframes should build on the work of your paper prototypes

- The only functionality that needs to work in your wireframe is navigation.

Wireframe

- Your goal is to get your users to provide you feedback on the aesthetics and placement of things on the screen.

- You should consider AODA (Accessibility for Ontarians with Disabilities Act) when designing your wireframe.

Comparing Prototypes

- Low fidelity (lo-fi)
  - paper prototyping

- High fidelity (hi-fi)
  - wireframing

Comparing Prototypes

- Time

- Quality of Feedback

- Changeability

- Feature Creep

- Program Bugs

- Users as Co-Designers

- Timing Studies

- Animation

Comparing Prototypes

- Time
    - LoFi is quick
    - HiFi is <u>not</u> quick

Comparing Prototypes

- Quality of feedback
  - LoFi is great. Users focus on interactions, functionality, and process flow.
  - HiFi focuses on finishing touches (colour, fonts, alignments, etc.)

Comparing Prototypes

- Changeability
    - LoFi invites change, and changes are made on the fly
    - HiFi isn't. Developers are resistant to change because so much time has been invested already

Comparing Prototypes

- Feature Creep
    - LoFi - development is still in design, so good chance of creep, but not finicky changes
    - HiFi looks complete, so finicky features are requested because you look finished & have time

Comparing Prototypes

- Program Bugs
    - LoFi - none
    - HiFi - can cause session to end, leaving you with nothing to show, and could end the project

Comparing Prototypes

- Users as Co-Designers
    - LoFi - invites suggestions and improvements
    - HiFi - less changes will be asked because things look "complete". Less ownership and buy-in.

Comparing Prototypes

- Timing Studies
    - LoFi - can't be done
    - HiFi - can be done; provides a rough estimate of "time on task"

Comparing Prototypes

- Animation
  - LoFi - can't be done
  - HiFi - can be done; but complicated

Wireframe activity

- Pick a web application that you're familiar with (e.g., Courselink, Webadvisor)
- Sketch a wireframe and discuss with your team
  - Structural elements
    - Functionality of individual elements
  - Layout
  - Navigation elements
  - Key scenarios