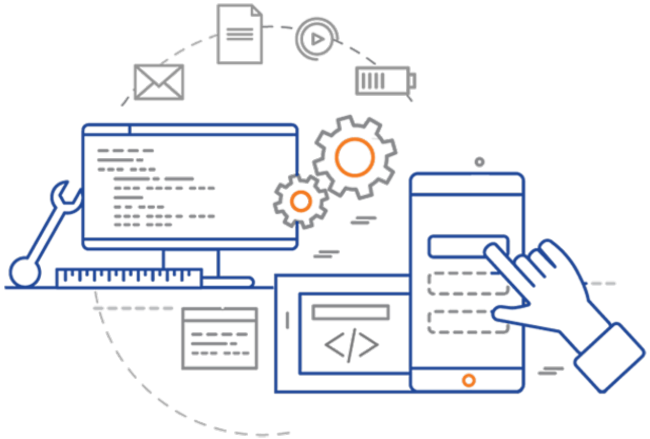


CS 398: Application Development

Week 04 Lecture: Analysis & Design 3

UX design; Design principles; SOLID principles.

In-Person Classes

We will be back in-person on Feb 7th.

- **Lectures will be in-person** i.e. no more MS teams calls like this.
- The **format is otherwise the same**: videos and slides will still be posted; in-class activities.

You are expected to continue to participate in team meets, MWF at scheduled times.

If your team approves, you may choose to call in through MS teams and participate remotely.

- We will allow for this in case you are ill, commuting etc.
- If you are physically in Waterloo, you should plan on attending.

If you cannot commit to being in-person, you should consider dropping the course.

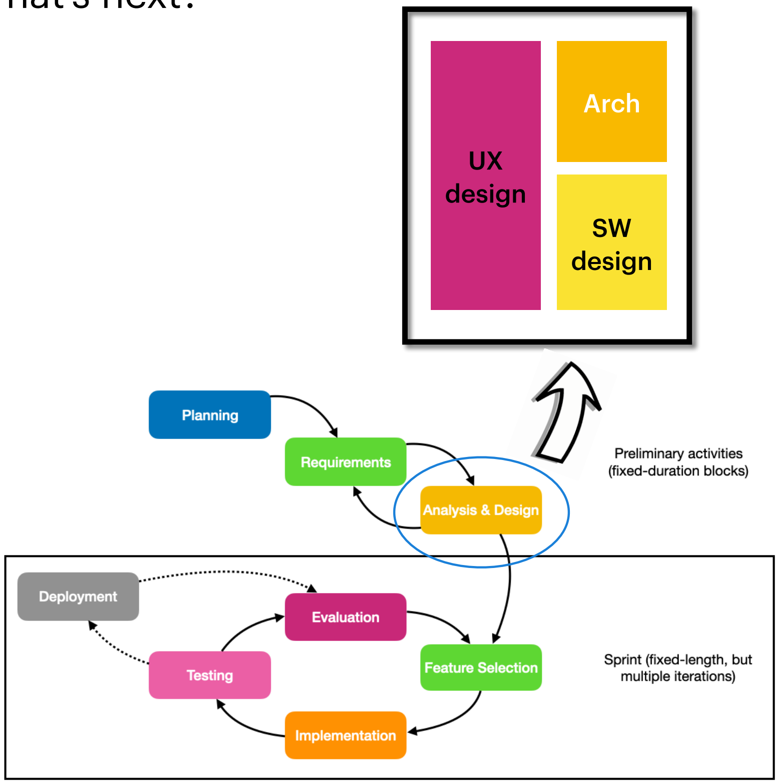
- The 100% refund date is Tues Jan 25 (i.e. tomorrow before the end of the day).

Phase 3: Analysis & Design

We've determined the high-level structure of our application. What's next?

1. We can consider the **UX design** of our system.
    - Mockups of critical screens.
    - Iterate on the user-interface design.
2. We can consider **software design** of our system:
    - What critical classes we might need.
    - How these classes should interrelate to one another.

UI prototypes are sometimes included in the requirements phase (i.e. iterating on user requirements). I prefer to think about the interface *after* making key technical decisions.



Software Development Lifecycle (SDLC)

This Week

Mon (today)

- Lectures: UX Design & Prototyping; Software design principles; SOLID principles.
- Activities: Sketching/prototyping

Wed

- Lectures: Design Patterns
- Activities: Consider if any patterns are relevant
  - Mention them in your presentation (2-3 sentences listing them and describing their relevance).
  - Add them to your system diagram.

Fri

- Lectures: software design video <— NOT testable, optional.
- Activities: Design review
  - 10 min presentation, followed by 5 min Q&A session.
  - Everyone must be present and everyone must present *something*.
  - See the website for your time. One of the course staff will call you on your team channel at that time.

Review

Incremental Design

How do you build compelling interfaces? You iterate on your designs!

1. Build a low-effort **prototype** of your application — a sketch, or mockup of what it will look like. If you have multiple windows, or screens, you will want to build prototypes of each screen.
2. Show the prototypes to users, and ask for **feedback**. Assess how easy they find it to use (not just appearance).
   - Helpful: “Here’s the screen that is intended to do X. How would you interact with this?”
   - Not helpful: “Here’s what I built. Do you like it?”
3. Use the feedback to iterate and improve on your design. Circle back to (1) until you are satisfied.

Avoid the temptation to “just code it”. A prototype deliberately represents a lower commitment (time, cost), encouraging you to discard or modify it as needed.

It can often take multiple iterations to get to a design that works well.

Low-Fidelity Prototypes

Low-fidelity prototypes are deliberately simple, low-tech, and represent a minimal investment.

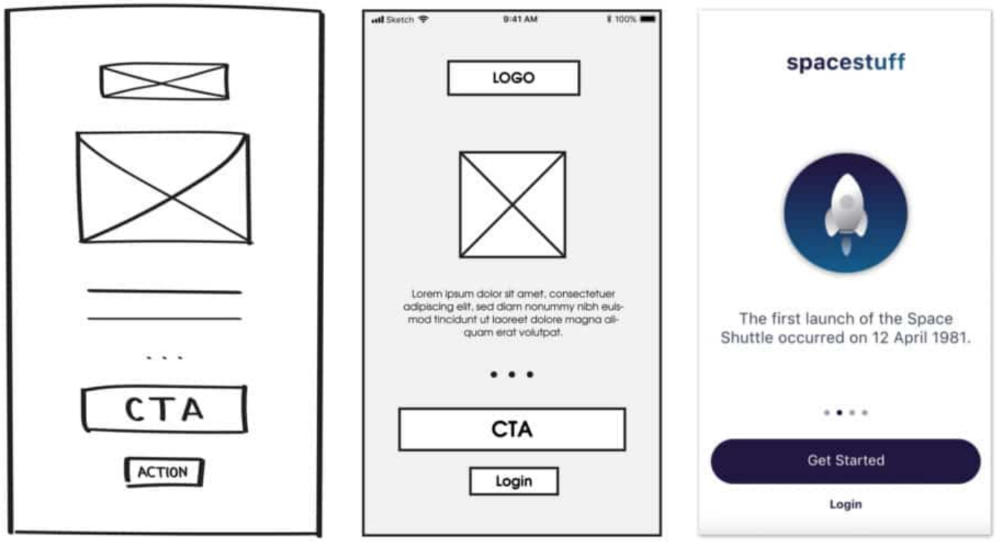
Building prototypes

- You can sketch something on paper.
- Many online tools help you build wireframe diagrams that you can show users.
- You can even make them semi-interactive to test progression through the interface.

What to prototype?

- Screens
- Interaction (through multiple screens)
- https://www.youtube.com/watch?v=yafaGNFu8Eg

These can get elaborate!


CTA
ACTION
Sketch
9:41 AM
100%
LOGO
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
CTA
Login
spacestuff
The first launch of the Space Shuttle occurred on 12 April 1981.
Get Started
Login

A low, medium and high-fidelity (final) version of a dialog.

Activities

TODO Today

✓ **Planning**

1. Create project plan

✓ **Requirements**

1. Pick users, (optional) create personas
2. Interview people that fall into your role
3. Identify requirements, (affinity diagram)
4. Document requirements in GitLab

Analysis & Design

1. Determine technical impact
2. Choose architectural style
3. System diagram
4. UI Mockup (Low-fidelity Prototype)
5. Design Patterns

Week 4 quiz is due Friday by 11:50 PM.
- All content included, with an emphasis on this week's lectures.
- Fri has an optional video - NOT testable.