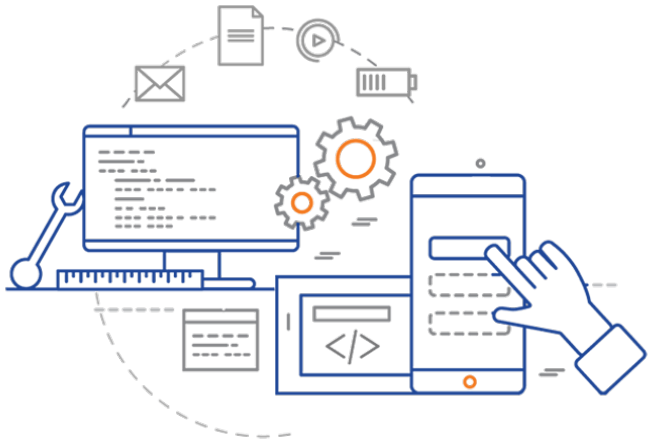

CS 398: Application Development

Week 06 Lecture: Sprint 1 Kickoff

Planning activities; Pair programming lectures

Welcome back!

Course staff introductions

- Licheng Zhang (l345zhan@), Masters Student, Teaching Assistant — MORNING
- Xiaoyan Xu (x439xu@), Masters Student, Teaching Assistant — ALWAYS HERE
- Yuan (Constant) Chen (y2238che@), PhD Student, Teaching Assistant — AFTERNOON

Licheng will be online, so feel free to message him or invite him to meetings in MS Teams.

This Week

Prior to this week, this course was structured around assigned activities. We're not doing that anymore.

From this point forward, you are self-directed - within the limits of a sprint.

Sprints all start with a kickoff meeting (today) and end with a demo on the last day.

	Mon	Wed	Fri
Week 1	Kickoff	(Standup)	(Standup)
Week 2	(Standup)	(Standup)	Demo

Sprint 1 Kickoff: Today
Sprint 1 Demo: Fri Feb 18th

Lecture videos are meant to help you during your sprints.

Week 6: Sprint 1 Kickoff

Goal: Sprint 1 Kickoff and Implemenation.

Mon:

- Before Lectures: pair programming video (slides) (notes)
- Lecture: slides
- Activities: Planning for Sprint 1.

Wed:

- Before Lectures: unit testing video (slides)
- Lecture: slides
- Activities: Daily standup. Activities planned by your team.

Fri:

- Before Lectures: refactoring video (slides)
- Lecture: slides
- Activities: Daily standup. Activities planned by your team.

Sprint 1 Kickoff

TODO

1. Decide what will be included in the Sprint
2. Assign work to everybody on the team
3. Do some work if you have time!

How to process the product backlog

1. First pass: review each item

- For each item, assign a priority relative to this sprint — high=yes, med=maybe, low=no
- For each high/med item, how much effort do you think it will take? — big/med/small effort

2. Second pass: assign items

- Sort the list from high-to-low priority.
- Work from the top-down, and assign items until the team runs out of capacity (i.e. time).

How do you prioritize or estimate?

What should be high priority?

- Items that logically need to be done before other features can be implemented. e.g. add a note.
- Domain classes e.g. note class, note container.
- Medium are things that are important but can wait. Low are things that you could skip completely.

How do you estimate time?

- Everything should be achievable in 1 day or less
  - If it's more than 1 day break it apart into smaller tasks.
  - Rate a task as small ("I can do this quickly"), medium ("some work but it can't take a whole day") and large ("I dunno, maybe a day?")
- Assume that if you don't know how to do it, it's "big".
- Use planning poker if you're stuck!

PLANNING POKER
1. Pick a number that represents your estimate (1-100 minutes or hours). Keep it hidden.
2. Everyone reveals estimates at the same time. High and low explain their reasoning.
3. Repeat until you converge on a number.

We've assigned work. Now what?

Your development model should look something like this:

1. Decide if you want to do paired programming.
   If so, think about physical/remote pairing. Consider using one of the online systems.

2. Pick a task from the list.

3. Create a branch in Git (a feature branch for your task).
   git checkout -b branch-name

4. Start coding!
   If you're doing pair programming, decide who is driving and who is navigating.
   If you're doing TDD, write unit tests first.