

Lecture 1: Course overview

About the course

Purpose: deliver quality software when the scope is larger and the stakes are higher

Themes

- Engineering principles
- Management & teamwork
- Design
- Construction & maintenance
- Quality
- Delivery
- Professionalism

Example success and failures

- Linux kernel (1991-present)
- SQLite (2000-present)
- Therac-25 (1982)
- Ariane 5 (1996)
- Boeing 737 MAX (2017)

Breakout 1: What is software engineering?

1. Take note of your breakout room number
2. Introduce yourselves to one another
 - a. Names:
 - b. Years/programs:

- c. What kinds of software you hope to write:
3. Select a recorder, responsible for writing down the results of your discussion
4. Discuss: What makes “software engineering” different from “programming”?

Syllabus

See website.

Poll

Which programming language would you (personally) be most comfortable using to implement a *large, reliable, maintainable* application?

Python, Java, OCaml, C++, Go, Rust, JavaScript

References

See website.

Enrollment

Send questions to cs-course-enroll@cornell.edu.

The project

Teams

- Teams of 6-8 students
- Check for compatible schedules and work styles
- Try to include members with frontend experience
- May form a complete or partial team around a project on your own, or may register to be matched with a team (surveys will be posted to Canvas this weekend)

CS 5150 SP22

- Teams should be formed by Feb 3
- Teamwork & collaboration subject of Lecture 3

Projects

Course staff will sponsor projects related to code review

- Demo: Gerrit code review tool

Teams may petition for an external project

- Client may not be a student
- Must involve existing, active codebase
- Submit survey “petition for external project” (check Canvas; coming soon)

Timeline: five 3-week sessions

Deliverables

- Session reports
 - Plan
 - Accomplishments, setbacks, discovered work
 - Peer feedback
- Work log
- Design documentation
- Test plans and reports
- Coverage analysis
- Requirements
- Code and code reviews
- User documentation

Breakout 2: Software engineering slang

1. Select a new recorder, responsible for writing down your definitions
2. Define the following terms in a software engineering context (don't just look them up; guessing and humor are okay):
 - a. Bike-shedding
 - b. Yak shaving
 - c. Dogfooding
 - d. Greenfield
 - e. MVP
 - f. DevOps

History

- Flow charts
- Structured programming
- Software crisis
- NATO software engineering conferences (1968-1969)
- Incremental improvements; no silver bullet

Demo: Counting lines of code

- Install ``tokei``
- ``git clone https://github.com/sxs-collaboration/spectre.git``
- ``tokei``
- ``tokei src`; `tokei tests``

Assignments

- Read about FAA Advanced Automation System (before Lecture 2)

CS 5150 SP22

- Count lines of code (due before Lecture 2; automatic extension if not yet enrolled on Canvas)
- Start forming teams around projects