

# CSE 403

Software Engineering

Spring 2025

## Course introduction

### Today

- The CSE 403 team
- Logistics and resources
- What is Software Engineering
- Course overview and expectations

### The CSE 403 team

#### Instructor

- René Just ([rjust@cs.washington.edu](mailto:rjust@cs.washington.edu))
- Office hours: After class and by appointment

#### Teaching assistants/project managers

- Afuza Afuzarahman
- Arnavi Mahendra Chheda
- Medha Gupta
- Melanie Kneitmix
- Connor Nicholas Reinholdtsen
- David Song

### Logistics: meetings

- **Lectures:** M/W/F 12:30pm – 1:20pm (G10)
- **Team meetings:** Tue 1:30pm – 2:20pm (G10)
- **Project meetings:** Thu 1:30pm – 2:20pm (G10)

Until 04/08 use Tue/Thu time to work on your project proposal with your assigned partner.

## Logistics: resources

- **Course website:**  
<https://homes.cs.washington.edu/~rjust/courses/CSE403> ([cs.uw.edu/403](https://cs.uw.edu/403))
- Submission of assignments via **Canvas**:  
<https://canvas.uw.edu>
- Project discussions on **Slack**:  
<https://cse403-sp25.slack.com>

## Logistics: communication

### Communication guidelines

- We use Slack for all **non-sensitive** project communication.
- See the [Slack guidelines](#) for this course.

### Resources

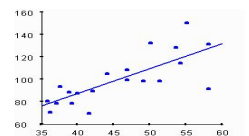
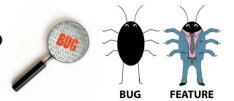
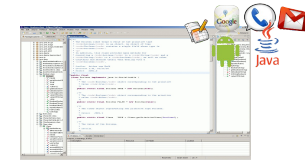
- The go-to page for this course is the [course web site](#).
- All relevant information is on the website, or linked from it.
- Canvas for assignments and non-public materials.

## Today

- The CSE 403 team
- Logistics and Background
- What is Software Engineering
- Course overview and expectations

## What is Software Engineering?

- Developing in an IDE and software ecosystem?
- Debugging and maintaining a software system?
- Deploying and running a software system?
- Empirically evaluating a software system?
- Writing (design) docs?



**All of the above and much more!**

# What is Software Engineering?

## More than just writing code

The complete process of specifying, designing, developing, analyzing, deploying, and maintaining a software system.

- Common Software Engineering tasks include:
  - Requirements engineering
  - Specification writing and documentation
  - Software architecture and design
  - Programming** Just one out of many important tasks!
  - Software testing and debugging
  - Maintenance and refactoring

# Why is Software Engineering important?

## Software is eating the world!



# Why is Software Engineering important?

## Software is eating the world!



# Summary: Software Engineering

## What is Software Engineering?

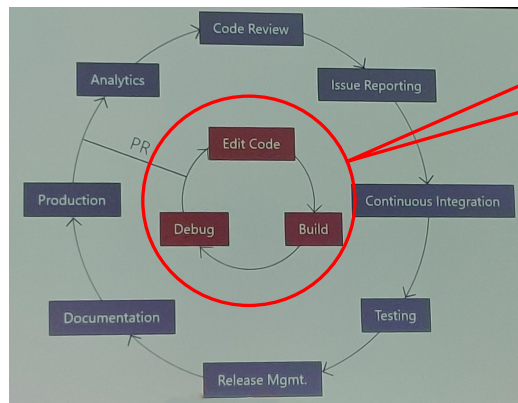
- The complete process of specifying, designing, developing, analyzing, and maintaining a software system.

## Why is it important?

- Decomposes a complex engineering problem.
- Organizes processes and effort.
- Improves software reliability.
- Improves developer productivity.

Does GenAI render Software Engineering obsolete?

# The Role of Software Engineering in Practice



Intro-level courses focus on the inner loop.

(Engineering workflow at Microsoft, Big Code summit 2019)

CSE 403 largely focuses on the outer loop.

## Today

- The CSE 403 team
- Logistics and Background
- What is Software Engineering
- Course overview and expectations

## Course overview: grading

Date	Topic	Materials	Readings	Assignments
03/01	Introduction			
04/01	Project proposal work		• Joel Test	Project proposal (due 04/08)
04/02	The Joel Test		• Joel Test: 20 years later	
04/03	Project proposal work		• What is Agile?	
04/04	Software development life cycle		• Agile vs. Traditional	
			• Test-driven development	
04/07	Teams and Scrum			Team setup (due 04/15)
04/08	Project proposal work			
04/09	Proposal pitches			
04/10	Proposal pitches			
04/11	Proposal pitches			
04/14	Requirements			Requirements (due 04/22)
04/15	Team meeting			
04/16	Requirements			
04/17	Project meeting			
04/18	Data modeling			
04/21	Architecture			Architecture & Design (due 04/28)
04/22	Team meeting			
04/23	Design			
04/24	Project meeting			
04/25	Version control and Git			
04/28	Build systems			Testing & CI (due 05/05)
04/29	Team meeting			
04/30	Testing and CI			
05/01	Project meeting			
05/02	In-class exercise (Git)			
05/05	Coverage-based testing			Data Release (due 05/13)
05/06	Team meeting			
05/07	Mutation-based testing			
05/08	Project meeting			
05/09	In-class exercise (Code defending)			
05/12	Hack day			Refinement (due 05/20)
05/13	Team meeting			
05/14	Code review			
05/15	Project meeting			
05/16	In-class exercise (Testing)			
05/19	Program analysis			Peer Review (due 05/27)
05/20	Team meeting			
05/21	Debugging			
05/22	Project meeting			
05/23	In-class exercise (Debugging)			
05/26	No class (Holiday)			
05/27	Team meeting			Final Release (due 06/03)
05/28	Fault localization			
05/29	Project meeting			
05/30	In-class exercise (Fault localization)			
06/02	Hack day			Individual reflection (due 06/10)
06/03	Team meeting			
06/04	Advanced program analysis			
06/05	Project meeting			
06/06	Optional in-class exercise			

### Grading

- 55%: Course project
  - 70% project milestones
  - 30% final project review
- 35%: In-class exercises and individual assignments
- 10%: Participation
  - Engagement in project meetings
  - In-class discussions and activities (polls, small-group activities, etc.)
  - Slack contributions
- **No final exam!**

## Course overview: workload

Date	Topic	Materials	Readings	Assignments
03/01	Introduction			
04/01	Project proposal work		• Joel Test	Project proposal (due 04/08)
04/02	The Joel Test		• Joel Test: 20 years later	
04/03	Project proposal work		• What is Agile?	
04/04	Software development life cycle		• Agile vs. Traditional	
			• Test-driven development	
04/07	Teams and Scrum			Team setup (due 04/15)
04/08	Project proposal work			
04/09	Proposal pitches			
04/10	Proposal pitches			
04/11	Proposal pitches			
04/14	Requirements			Requirements (due 04/22)
04/15	Team meeting			
04/16	Requirements			
04/17	Project meeting			
04/18	Data modeling			
04/21	Architecture			Architecture & Design (due 04/28)
04/22	Team meeting			
04/23	Design			
04/24	Project meeting			
04/25	Version control and Git			
04/28	Build systems			Testing & CI (due 05/05)
04/29	Team meeting			
04/30	Testing and CI			
05/01	Project meeting			
05/02	In-class exercise (Git)			
05/05	Coverage-based testing			Data Release (due 05/13)
05/06	Team meeting			
05/07	Mutation-based testing			
05/08	Project meeting			
05/09	In-class exercise (Code defending)			
05/12	Hack day			Refinement (due 05/20)
05/13	Team meeting			
05/14	Code review			
05/15	Project meeting			
05/16	In-class exercise (Testing)			
05/19	Program analysis			Peer Review (due 05/27)
05/20	Team meeting			
05/21	Debugging			
05/22	Project meeting			
05/23	In-class exercise (Debugging)			
05/26	No class (Holiday)			
05/27	Team meeting			Final Release (due 06/03)
05/28	Fault localization			
05/29	Project meeting			
05/30	In-class exercise (Fault localization)			
06/02	Hack day			Individual reflection (due 06/10)
06/03	Team meeting			
06/04	Advanced program analysis			
06/05	Project meeting			
06/06	Optional in-class exercise			

### Grading

- 55%: Course project
- 35%: In-class exercises and individual assignments
- 10%: Participation
- **No final exam!**

### Workload

- One project assignment each week

# Course overview: workload

Date	Topic	Materials	Readings	Assignments
04/01	Introduction			
04/01	Project proposal work		• Joel Test	Project proposal (due 04/08)
04/02	The Joel Test		• Joel Test: 20 years later	
04/03	Project proposal work		• What is Agile?	
04/04	Software development life cycle		• Agile vs. Traditional	
			• Test-driven development	
04/07	Teams and Scrum			
04/08	Project proposal work			Team setup (due 04/15)
04/09	Proposal pitches			
04/10	Proposal pitches			
04/11	Proposal pitches			
04/14	Requirements			
04/15	Team meeting			
04/16	Requirements			Requirements (due 04/22)
04/17	Project meeting			
04/18	Data modelling			
04/21	Architecture			
04/22	Team meeting			
04/23	Design			Architecture & Design (due 04/28)
04/24	Project meeting			
04/25	Version control and Git			
04/28	Build systems			
04/29	Team meeting			
04/30	Testing and CI			Testing & CI (due 05/06)
05/01	Project meeting			
05/02	In-class exercise (Git)			
05/05	Coverage-based testing			
05/06	Team meeting			
05/07	Mutation-based testing			
05/08	Project meeting			Data Release (due 05/13)
05/09	In-class exercise (Code defensively)			
05/12	Hack day			
05/13	Team meeting			
05/14	Code review			Refinement (due 05/20)
05/15	Project meeting			
05/16	In-class exercise (Testing)			
05/19	Program analysis			
05/20	Team meeting			
05/21	Debugging			Peer Review (due 05/27)
05/22	Project meeting			
05/23	In-class exercise (Debugging)			
05/26	No class (Holiday)			
05/27	Team meeting			
05/28	Fault localization			
05/29	Project meeting			Final Release (due 06/03)
05/30	In-class exercise (Fault localization)			
06/02	Hack day			
06/03	Team meeting			
06/04	Advanced program analysis			Individual reflection (due 06/10)
06/05	Project meeting			
06/06	Optional in-class exercise			

## Grading

- 55%: Course project
- 35%: In-class exercises and individual assignments
- 10%: Participation
- No final exam!

## Workload

- One project assignment each week
- 5 (+1 optional) in-class exercises

# Course overview: workload

Date	Topic	Materials	Readings	Assignments
04/01	Introduction			
04/01	Project proposal work		• Joel Test	Project proposal (due 04/08)
04/02	The Joel Test		• Joel Test: 20 years later	
04/03	Project proposal work		• What is Agile?	
04/04	Software development life cycle		• Agile vs. Traditional	
			• Test-driven development	
04/07	Teams and Scrum			
04/08	Project proposal work			Team setup (due 04/15)
04/09	Proposal pitches			
04/10	Proposal pitches			
04/11	Proposal pitches			
04/14	Requirements			
04/15	Team meeting			
04/16	Requirements			Requirements (due 04/22)
04/17	Project meeting			
04/18	Data modelling			
04/21	Architecture			
04/22	Team meeting			
04/23	Design			Architecture & Design (due 04/28)
04/24	Project meeting			
04/25	Version control and Git			
04/28	Build systems			
04/29	Team meeting			
04/30	Testing and CI			Testing & CI (due 05/06)
05/01	Project meeting			
05/02	In-class exercise (Git)			
05/05	Coverage-based testing			
05/06	Team meeting			
05/07	Mutation-based testing			
05/08	Project meeting			Data Release (due 05/13)
05/09	In-class exercise (Code defensively)			
05/12	Hack day			
05/13	Team meeting			
05/14	Code review			Refinement (due 05/20)
05/15	Project meeting			
05/16	In-class exercise (Testing)			
05/19	Program analysis			
05/20	Team meeting			
05/21	Debugging			Peer Review (due 05/27)
05/22	Project meeting			
05/23	In-class exercise (Debugging)			
05/26	No class (Holiday)			
05/27	Team meeting			
05/28	Fault localization			
05/29	Project meeting			Final Release (due 06/03)
05/30	In-class exercise (Fault localization)			
06/02	Hack day			
06/03	Team meeting			
06/04	Advanced program analysis			Individual reflection (due 06/10)
06/05	Project meeting			
06/06	Optional in-class exercise			

## Grading

- 55%: Course project
- 35%: In-class exercises and individual assignments
- 10%: Participation
- No final exam!

## Workload

- One project assignment each week
- 5 (+1 optional) in-class exercises
- Extra time allocated for crunch time

# Course overview: topics

Date	Topic	Materials	Readings	Assignments
04/01	Introduction			
04/01	Project proposal work		• Joel Test	Project proposal (due 04/08)
04/02	The Joel Test		• Joel Test: 20 years later	
04/03	Project proposal work		• What is Agile?	
04/04	Software development life cycle		• Agile vs. Traditional	
			• Test-driven development	
04/07	Teams and Scrum			
04/08	Project proposal work			Team setup (due 04/15)
04/09	Proposal pitches			
04/10	Proposal pitches			
04/11	Proposal pitches			
04/14	Requirements			
04/15	Team meeting			
04/16	Requirements			Requirements (due 04/22)
04/17	Project meeting			
04/18	Data modelling			
04/21	Architecture			
04/22	Team meeting			
04/23	Design			Architecture & Design (due 04/28)
04/24	Project meeting			
04/25	Version control and Git			
04/28	Build systems			
04/29	Team meeting			
04/30	Testing and CI			Testing & CI (due 05/06)
05/01	Project meeting			
05/02	In-class exercise (Git)			
05/05	Coverage-based testing			
05/06	Team meeting			
05/07	Mutation-based testing			
05/08	Project meeting			Data Release (due 05/13)
05/09	In-class exercise (Code defensively)			
05/12	Hack day			
05/13	Team meeting			
05/14	Code review			Refinement (due 05/20)
05/15	Project meeting			
05/16	In-class exercise (Testing)			
05/19	Program analysis			
05/20	Team meeting			
05/21	Debugging			Peer Review (due 05/27)
05/22	Project meeting			
05/23	In-class exercise (Debugging)			
05/26	No class (Holiday)			
05/27	Team meeting			
05/28	Fault localization			
05/29	Project meeting			Final Release (due 06/03)
05/30	In-class exercise (Fault localization)			
06/02	Hack day			
06/03	Team meeting			
06/04	Advanced program analysis			Individual reflection (due 06/10)
06/05	Project meeting			
06/06	Optional in-class exercise			

- Software processes, requirements, and specification
  - Different software development processes.
  - Precise writing (requirements and specifications).

# Course overview: topics

Date	Topic	Materials	Readings	Assignments
04/01	Introduction			
04/01	Project proposal work		• Joel Test	Project proposal (due 04/08)
04/02	The Joel Test		• Joel Test: 20 years later	
04/03	Project proposal work		• What is Agile?	
04/04	Software development life cycle		• Agile vs. Traditional	
			• Test-driven development	
04/07	Teams and Scrum			
04/08	Project proposal work			Team setup (due 04/15)
04/09	Proposal pitches			
04/10	Proposal pitches			
04/11	Proposal pitches			
04/14	Requirements			
04/15	Team meeting			
04/16	Requirements			Requirements (due 04/22)
04/17	Project meeting			
04/18	Data modelling			
04/21	Architecture			
04/22	Team meeting			
04/23	Design			Architecture & Design (due 04/28)
04/24	Project meeting			
04/25	Version control and Git			
04/28	Build systems			
04/29	Team meeting			
04/30	Testing and CI			Testing & CI (due 05/06)
05/01	Project meeting			
05/02	In-class exercise (Git)			
05/05	Coverage-based testing			
05/06	Team meeting			
05/07	Mutation-based testing			
05/08	Project meeting			Data Release (due 05/13)
05/09	In-class exercise (Code defensively)			
05/12	Hack day			
05/13	Team meeting			
05/14	Code review			Refinement (due 05/20)
05/15	Project meeting			
05/16	In-class exercise (Testing)			
05/19	Program analysis			
05/20	Team meeting			
05/21	Debugging			Peer Review (due 05/27)
05/22	Project meeting			
05/23	In-class exercise (Debugging)			
05/26	No class (Holiday)			
05/27	Team meeting			
05/28	Fault localization			
05/29	Project meeting			Final Release (due 06/03)
05/30	In-class exercise (Fault localization)			
06/02	Hack day			
06/03	Team meeting			
06/04	Advanced program analysis			Individual reflection (due 06/10)
06/05	Project meeting			
06/06	Optional in-class exercise			

- Software processes, requirements, and specification
  - Different software development processes.
  - Precise writing (requirements and specifications).
- Software development
  - Decompose a complex problem and build abstractions.
  - Improve your coding skills.
  - Effectively use version control, build systems, and code review.
  - Continuous integration (CI).

# Course overview: topics

Course material				
Date	Topic	Materials	Readings	Assignments
03/01	Introduction			
04/01	Project proposal work		• Joel Test	Project proposal (due 04/08)
04/02	The Joel Test		• Joel Test: 20 years later	
04/03	Project proposal work			
04/04	Software development life cycle		• What is Agile? • Agile vs. Traditional • Test-driven development	
04/07	Teams and Scrum			
04/08	Project proposal work			Team setup (due 04/15)
04/09	Proposal pitches			
04/10	Proposal pitches			
04/11	Proposal pitches			
04/14	Requirements			
04/15	Team meeting			
04/16	Requirements			Requirements (due 04/22)
04/17	Project meeting			
04/18	Data modelling			
04/21	Architecture			
04/22	Team meeting			
04/23	Design			Architecture & Design (due 04/28)
04/24	Project meeting			
04/25	Version control and Git			
04/28	Build systems			
04/29	Team meeting			
04/30	Testing and CI			Testing & CI (due 05/06)
05/01	Project meeting			
05/02	In-class exercise (Git)			
05/05	Coverage-based testing			
05/06	Team meeting			
05/07	Mutation-based testing			Data Release (due 05/13)
05/08	Project meeting			
05/09	In-class exercise (Code defenders)			
05/12	Hack day			
05/13	Team meeting			
05/14	Code review			Refinement (due 05/20)
05/15	Project meeting			
05/16	In-class exercise (Testing)			
05/19	Program analysis			
05/20	Team meeting			
05/21	Debugging			Peer Review (due 05/27)
05/22	Project meeting			
05/23	In-class exercise (Debugging)			
05/26	No class (Holiday)			
05/27	Team meeting			
05/28	Fault localisation			
05/29	Project meeting			Final Release (due 06/03)
05/30	In-class exercise (Fault localisation)			
06/02	Hack day			
06/03	Team meeting			
06/04	Advanced program analysis			Individual reflection (due 06/10)
06/05	Project meeting			
06/06	Optional in-class exercise			

- **Software processes, requirements, and specification**
  - Different software development processes.
  - Precise writing (requirements and specifications).
- **Software development**
  - Decompose a complex problem and build abstractions.
  - Improve your coding skills.
  - Effectively use version control, build systems, and code review.
  - Continuous integration (CI).
- **Software testing and debugging**
  - Write effective (unit) tests.
  - Hands-on experience, using testing and debugging techniques.
  - (Advanced) program analysis.

# Course overview: course project

Course material				
Date	Topic	Materials	Readings	Assignments
03/01	Introduction			
04/01	Project proposal work		• Joel Test	Project proposal (due 04/08)
04/02	The Joel Test		• Joel Test: 20 years later	
04/03	Project proposal work			
04/04	Software development life cycle		• What is Agile? • Agile vs. Traditional • Test-driven development	
04/07	Teams and Scrum			
04/08	Project proposal work			Team setup (due 04/15)
04/09	Proposal pitches			
04/10	Proposal pitches			
04/11	Proposal pitches			
04/14	Requirements			
04/15	Team meeting			
04/16	Requirements			Requirements (due 04/22)
04/17	Project meeting			
04/18	Data modelling			
04/21	Architecture			
04/22	Team meeting			
04/23	Design			Architecture & Design (due 04/28)
04/24	Project meeting			
04/25	Version control and Git			
04/28	Build systems			
04/29	Team meeting			
04/30	Testing and CI			Testing & CI (due 05/06)
05/01	Project meeting			
05/02	In-class exercise (Git)			
05/05	Coverage-based testing			
05/06	Team meeting			
05/07	Mutation-based testing			Data Release (due 05/13)
05/08	Project meeting			
05/09	In-class exercise (Code defenders)			
05/12	Hack day			
05/13	Team meeting			
05/14	Code review			Refinement (due 05/20)
05/15	Project meeting			
05/16	In-class exercise (Testing)			
05/19	Program analysis			
05/20	Team meeting			
05/21	Debugging			Peer Review (due 05/27)
05/22	Project meeting			
05/23	In-class exercise (Debugging)			
05/26	No class (Holiday)			
05/27	Team meeting			
05/28	Fault localisation			
05/29	Project meeting			Final Release (due 06/03)
05/30	In-class exercise (Fault localisation)			
06/02	Hack day			
06/03	Team meeting			
06/04	Advanced program analysis			Individual reflection (due 06/10)
06/05	Project meeting			
06/06	Optional in-class exercise			

- **Software processes, requirements, and specification**
  - Different software development processes.
  - Precise writing (requirements and specifications).
- **Software development**
  - Decompose a complex problem and build abstractions.
  - Improve your coding skills.
  - Effectively use version control, build systems, and code review.
  - Continuous integration (CI).
- **Software testing and debugging**
  - Write effective (unit) tests.
  - Hands-on experience, using testing and debugging techniques.
  - (Advanced) program analysis.
- **Course project**
  - Apply all of the above in a group project.

## Course project overview

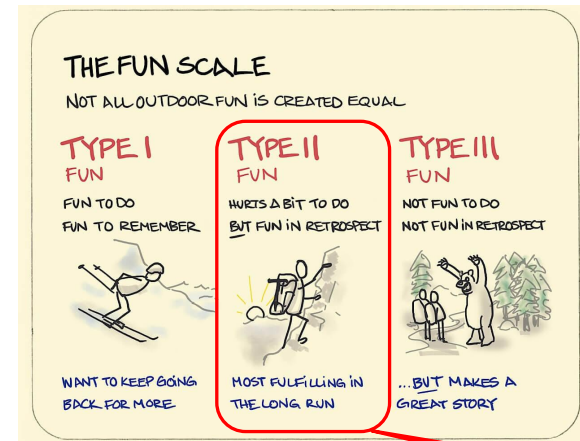
## Course project proposals

## Course project categories

### Example categories

- Productivity and convenience apps
- Optimization problems and data science
- Gaming and making
- Extensions to open-source software
- Software Engineering research (prototypes)

## CSE 403 in one picture: mostly type II fun



Sweet spot for teaching

## Expectations

- Programming experience and familiarity with one programming language (Java, C++, ...).
- Active participation in discussions.
- Teamwork and communication (Slack).
- Reflecting on and improving submitted materials.

## CSE 403: challenges for students

### Team work

- Effective communication and coordination
- Different backgrounds, skills, and incentives

### Complexity

- Tooling and technology stacks
- Scale of code base

### Uncertainty

- No simple check-box grading
- Focus on trade-offs, decisions, and justifications



# CSE 403: challenges for students and staff

## The Week-1 rush



## Enrollment

- 2020: 40 students (2 TAs)
- 2021: 85 students (5 TAs)
- 2022: 110 students (6 TAs)
- 2023: 82 students (5 TAs)
- 2025: 100 students (6 TAs)

## Lecture time (12:30)



## Time

- Project duration: 9 weeks
- Lecture time: 50 minutes
- Quick turnaround times (milestones and grading)

## What's next?

- *Tue: Work on project proposal (pre-assigned groups)*
- *Wed: The Joel Test (or why you really should take 403)*
- *Thu: Work on project proposal (pre-assigned groups)*
- *Fri: SDLC: Software Development Life Cycle*