CSE 403

Software Engineering

Course introduction

Today

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

The CSE 403 team

Instructor

- Michael Ernst; office hours: after class and by appointment
- Best email: cse403-staff@cs; you may also use mernst@cs.

TAs

- Saket Gollapudi
- Jason Hoffman
- Allan Ji
- Melanie Kneitmix
- Mitchell Levy
- Mingyuan Zhong

Your TA has multiple roles:

- Venture capitalist who expects results
- Manager who helps when you have difficulty
- Grader

5 meetings per week

- **Lectures**: MWF; Friday is often an in-class activity
- **Team meeting**: Tuesday
- **Meet with your TA**: Thursday

The first week and a half or so have a different schedule.

This Thursday: work on project proposal with your assigned partner.

Logistics: resources

- **Course website**:
  https://homes.cs.washington.edu/~rjust/courses/CSE403 (cs.uw.edu/403)

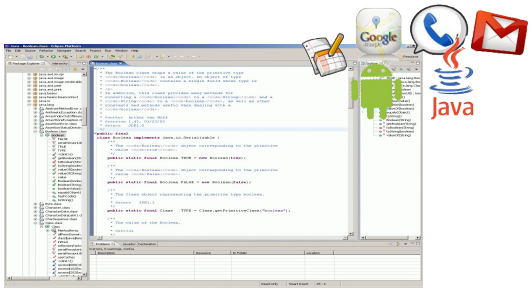
  All relevant information is on the website, or linked from it.
- Submit assignments via **Canvas**:
  https://canvas.uw.edu

Today

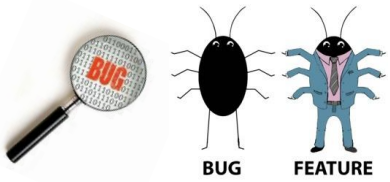
- The CSE 403 team
- Logistics and Background
- What is Software Engineering
- Course overview and expectations

What is Software Engineering?

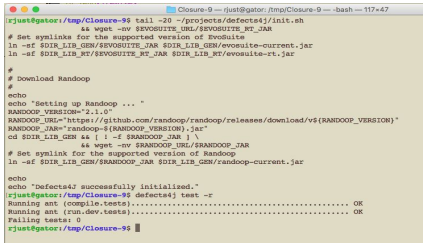
- Developing in an IDE
  and software ecosystem?



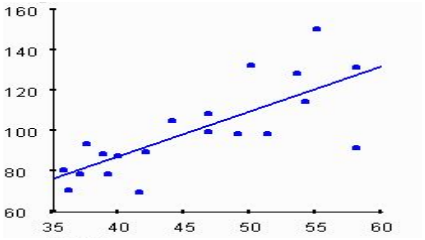
- Debugging and maintaining a software system?



- Deploying and running
  a software system?



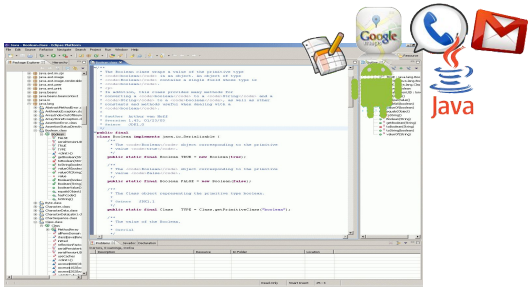
- Empirically evaluating a software system?



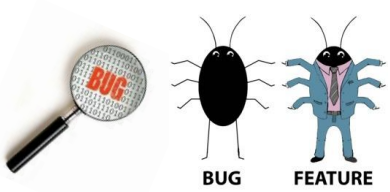
- Writing (design) docs?

What is Software Engineering?

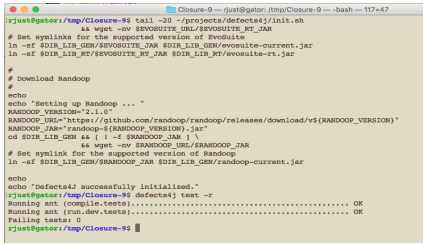
- Developing in an IDE and software ecosystem?



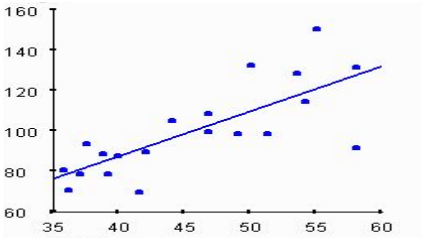
- Debugging and maintaining a software system?



- Deploying and running a software system?



- Empirically evaluating a software system?



- Writing (design) docs?

All of the above and much more!

Software Engineering is more than writing code

Software Engineering is the complete process of

specifying,
requirements engineering, specifications, documentation

designing, software architecture and design, UI

developing, programming (just one of many important tasks)

analyzing,
testing, debugging, linting, verification, performance engineering

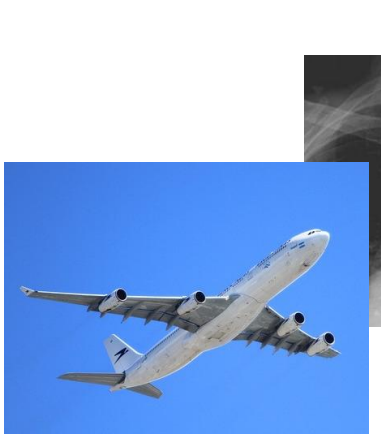
deploying,
DevOps, CI, packaging, operation, remote diagnostics, documentation, websites

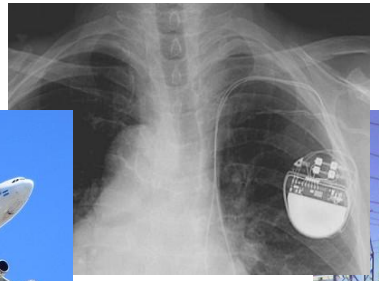
& maintaining refactoring, extensions, adaptation, issue tracking

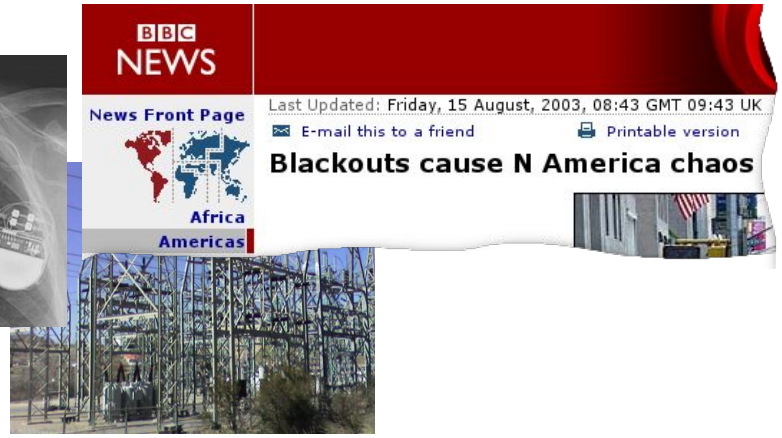
a software system.

Why is Software Engineering important?

Software is everywhere!

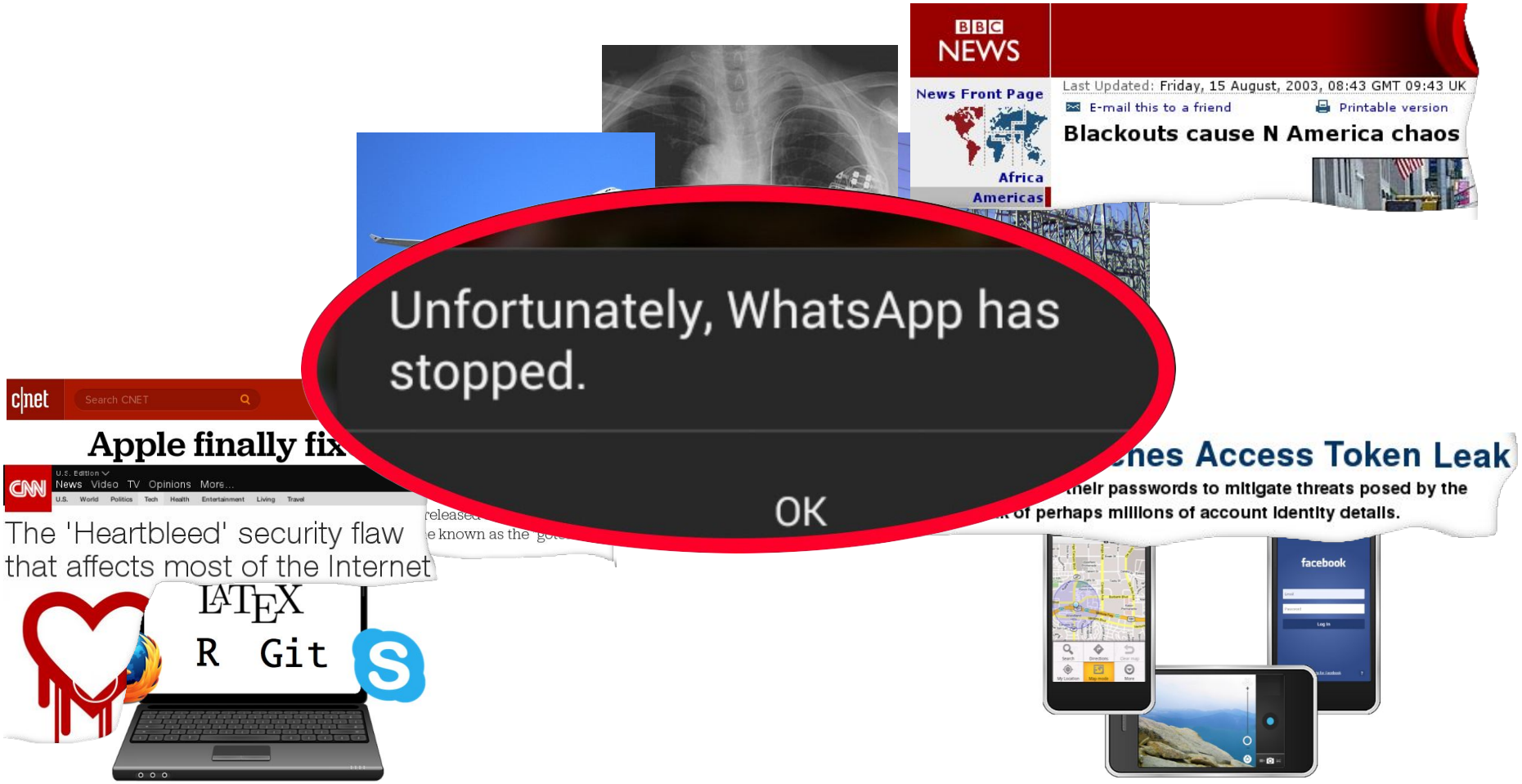






Why is Software Engineering important?

Software is everywhere!



Summary: Software Engineering

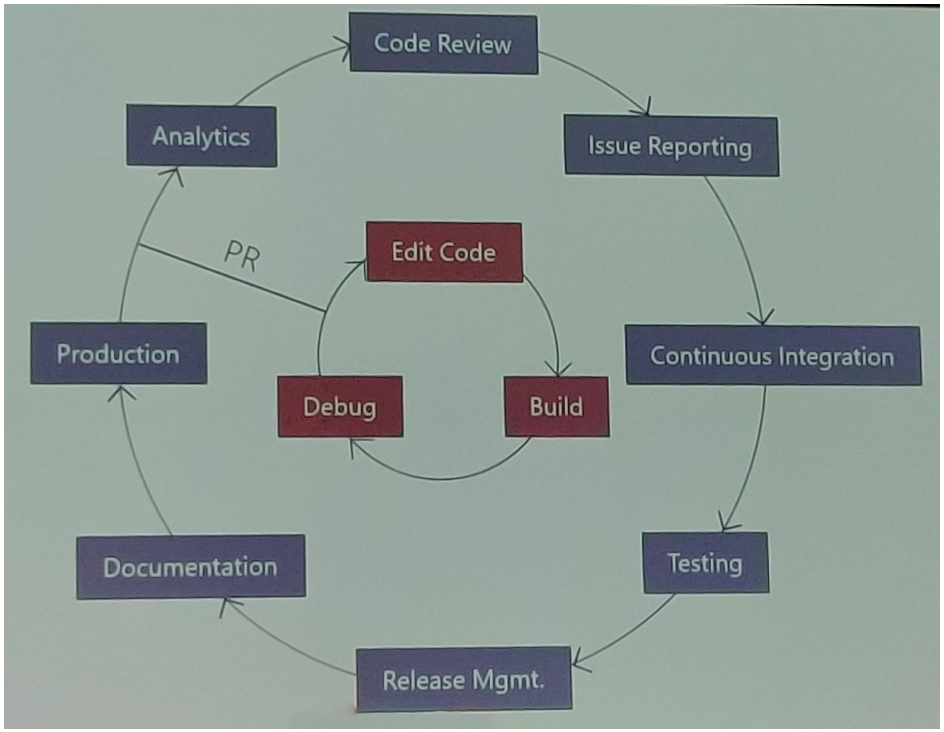
What is Software Engineering?

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

Why is it important?

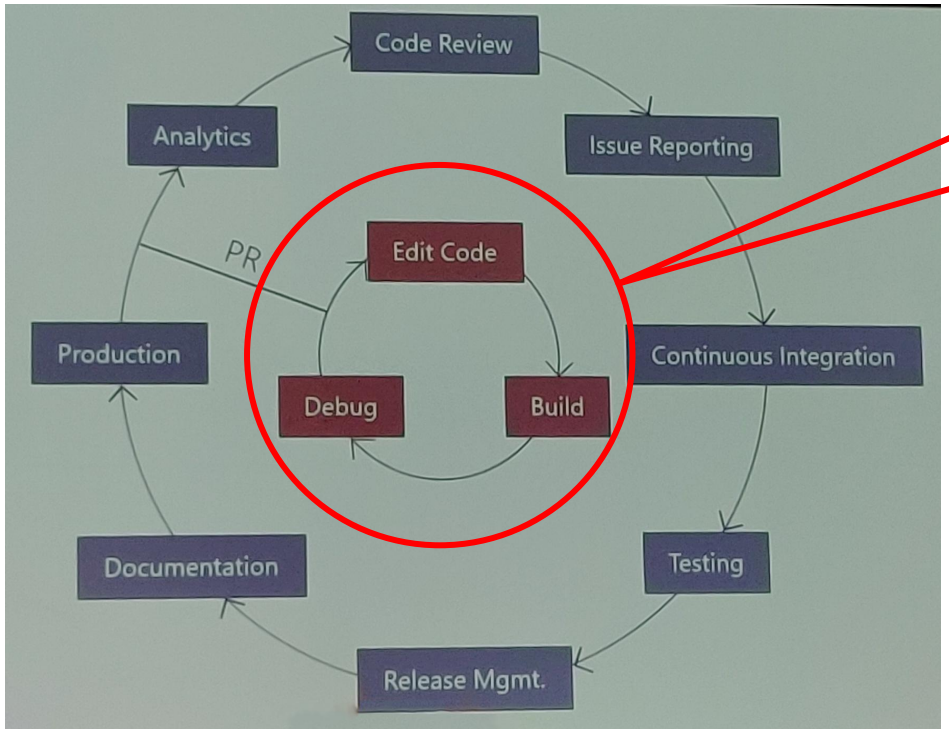
- Decomposes a complex engineering problem.
- Organizes processes and effort.
- Improves software reliability.
- Improves developer productivity.
- Leads to a successful product!
  Both technical and management contributions are essential.

The Role of Software Engineering in Practice



(Engineering workflow at Microsoft, Big Code summit 2019)

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.

What can you learn in CSE 403

- Learn best software development best practices
- Understand how software is produced – from conception to continuous development and release
- Develop skills to effectively collaborate with others towards a common delivery goal
- Experience the responsibilities, issues and tradeoffs involved in making decisions as software engineers

Much of the above is grounded by working with a team to incrementally deliver a real software product/service

Today

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

Course overview: grading

Course material

Date	Topic	Materials	Assignments
01/02	***No class (holiday)***		
01/03	***No section***		
01/04	Introduction		Project proposal (due 01/09)
01/05	*Project proposals*		
01/06	The Joel Test	Reading 1 and 2	
01/09	Software development life cycle		
01/10	*Project proposals*		
01/11	Requirements and Use cases		Requirements and policies (due 01/17)
01/12	*Project meeting*		
01/13	Teams and Scrum		
01/16	***No class (holiday)***		
01/17	*Team meeting*		
01/18	Version control and Git		Git setup (due 01/24)
01/19	*Project meeting*		
01/20	**In-class exercise (Git)**	Canvas	
01/23	Data modelling		
01/24	*Team meeting*		
01/25	Architecture		Architecture and Design (due 01/31)
01/26	*Project meeting*		
01/27	Design	Reading (Sections 1-6)	
01/30	Build systems		
01/31	*Team meeting*		
02/01	Testing and CI	Ant+GH Actions Gradle+Travis CI	Testing and CI (due 02/07)
02/02	*Project meeting*		
02/03	Code review	Tutorial video	
02/06	Coverage-based testing	Reading	
02/07	*Team meeting*		
02/08	Mutation-based testing	Reading 1 and 2	Beta release (due 02/14)
02/09	*Project meeting*		
02/10	**In-class exercise (Code defenders)**	Canvas	
02/13	***Hack day***		
02/14	*Team meeting*		
02/15	Reflection		Implementation and Documentation (due 02/21)
02/16	*Project meeting*		
02/17	**In-class exercise (Testing)**	Canvas	
02/20	***No class (holiday)***		
02/21	*Team meeting*		
02/22	Debugging	Reading	Peer review (due 02/28)
02/23	*Project meeting*		
02/24	**In-class exercise (Debugging)**	Canvas	
02/27	Program analysis		
02/28	*Team meeting*		
03/01	Fault localization		Final release (due 03/07)
03/02	*Project meeting*		
03/03	**In-class exercise (Fault localization)**	Canvas	
03/06	***Hack day***		
03/07	*Team meeting*		
03/08	Advanced program analysis		Reflection (due 03/14)
03/09	*Project meeting*		
03/10	Optional in-class exercise	Canvas	

Grading

- 55%: Course project
  - 70% project milestones
  - 30% final project review
- 20%: In-class exercises and individual assignments
- 15%: Midterm exam
- 10%: Participation
  - Engagement in project meetings
  - In-class discussions and activities (polls, small-group activities, etc.)
  - Slack contributions
- No final exam

Course overview: workload

Course material

Date	Topic	Materials	Assignments
01/02	***No class (holiday)***		
01/03	***No section***		
01/04	Introduction		Project proposal (due 01/09)
01/05	*Project proposals*		
01/06	The Joel Test	Reading 1 and 2	
01/09	Software development life cycle		
01/10	*Project proposals*		
01/11	Requirements and Use cases		Requirements and policies (due 01/17)
01/12	*Project meeting*		
01/13	Teams and Scrum		
01/16	***No class (holiday)***		
01/17	*Team meeting*		
01/18	Version control and Git		Git setup (due 01/24)
01/19	*Project meeting*		
01/20	**In-class exercise (Git)**	Canvas	
01/23	Data modelling		
01/24	*Team meeting*		
01/25	Architecture		Architecture and Design (due 01/31)
01/26	*Project meeting*		
01/27	Design	Reading (Sections 1-6)	
01/30	Build systems		
01/31	*Team meeting*		
02/01	Testing and CI	Ant+GH Actions Gradle+Travis CI	Testing and CI (due 02/07)
02/02	*Project meeting*		
02/03	Code review	Tutorial video	
02/06	Coverage-based testing	Reading	
02/07	*Team meeting*		
02/08	Mutation-based testing	Reading 1 and 2	Beta release (due 02/14)
02/09	*Project meeting*		
02/10	**In-class exercise (Code defenders)**	Canvas	
02/13	***Hack day***		
02/14	*Team meeting*		
02/15	Reflection		Implementation and Documentation (due 02/21)
02/16	*Project meeting*		
02/17	**In-class exercise (Testing)**	Canvas	
02/20	***No class (holiday)***		
02/21	*Team meeting*		
02/22	Debugging	Reading	Peer review (due 02/28)
02/23	*Project meeting*		
02/24	**In-class exercise (Debugging)**	Canvas	
02/27	Program analysis		
02/28	*Team meeting*		
03/01	Fault localization		Final release (due 03/07)
03/02	*Project meeting*		
03/03	**In-class exercise (Fault localization)**	Canvas	
03/06	***Hack day***		
03/07	*Team meeting*		
03/08	Advanced program analysis		Reflection (due 03/14)
03/09	*Project meeting*		
03/10	Optional in-class exercise	Canvas	

Grading

- 55%: Course project
- 20%: In-class exercises and individual assignments
- 15%: Mid-term exam
- 10%: Participation

Workload

- One project assignment each week

Course overview: workload

Course material

Date	Topic	Materials	Assignments
01/02	***No class (holiday)***		
01/03	***No section***		
01/04	Introduction		Project proposal (due 01/09)
01/05	*Project proposals*		
01/06	The Joel Test	Reading 1 and 2	
01/09	Software development life cycle		
01/10	*Project proposals*		
01/11	Requirements and Use cases		Requirements and policies (due 01/17)
01/12	*Project meeting*		
01/13	Teams and Scrum		
01/16	***No class (holiday)***		
01/17	*Team meeting*		
01/18	Version control and Git		Git setup (due 01/24)
01/19	*Project meeting*		
01/20	**In-class exercise (Git)**	Canvas	
01/23	Data modelling		
01/24	*Team meeting*		
01/25	Architecture		Architecture and Design (due 01/31)
01/26	*Project meeting*		
01/27	Design	Reading (Sections 1-6)	
01/30	Build systems		
01/31	*Team meeting*		
02/01	Testing and CI	Ant+GH Actions Gradle+Travis CI	Testing and CI (due 02/07)
02/02	*Project meeting*		
02/03	Code review	Tutorial video	
02/06	Coverage-based testing	Reading	
02/07	*Team meeting*		
02/08	Mutation-based testing	Reading 1 and 2	Beta release (due 02/14)
02/09	*Project meeting*		
02/10	**In-class exercise (Code defenders)**	Canvas	
02/13	***Hack day***		
02/14	*Team meeting*		
02/15	Reflection		Implementation and Documentation (due 02/21)
02/16	*Project meeting*		
02/17	**In-class exercise (Testing)**	Canvas	
02/20	***No class (holiday)***		
02/21	*Team meeting*		
02/22	Debugging	Reading	Peer review (due 02/28)
02/23	*Project meeting*		
02/24	**In-class exercise (Debugging)**	Canvas	
02/27	Program analysis		
02/28	*Team meeting*		
03/01	Fault localization		Final release (due 03/07)
03/02	*Project meeting*		
03/03	**In-class exercise (Fault localization)**	Canvas	
03/06	***Hack day***		
03/07	*Team meeting*		
03/08	Advanced program analysis		Reflection (due 03/14)
03/09	*Project meeting*		
03/10	Optional in-class exercise	Canvas	

Grading

- 55%: Course project
- 20%: In-class exercises and individual assignments
- 15%: Mid-term exam
- 10%: Participation

Workload

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

Course overview: workload

Course material

Date	Topic	Materials	Assignments
01/02	***No class (holiday)***		
01/03	***No section***		
01/04	Introduction		Project proposal (due 01/09)
01/05	*Project proposals*		
01/06	The Joel Test	Reading 1 and 2	
01/09	Software development life cycle		
01/10	*Project proposals*		
01/11	Requirements and Use cases		Requirements and policies (due 01/17)
01/12	*Project meeting*		
01/13	Teams and Scrum		
01/16	***No class (holiday)***		
01/17	*Team meeting*		
01/18	Version control and Git		Git setup (due 01/24)
01/19	*Project meeting*		
01/20	**In-class exercise (Git)**	Canvas	
01/23	Data modelling		
01/24	*Team meeting*		
01/25	Architecture		Architecture and Design (due 01/31)
01/26	*Project meeting*		
01/27	Design	Reading (Sections 1-6)	
01/30	Build systems		
01/31	*Team meeting*		
02/01	Testing and CI	Ant+GH Actions Gradle+Travis CI	Testing and CI (due 02/07)
02/02	*Project meeting*		
02/03	Code review	Tutorial video	
02/06	Coverage-based testing	Reading	
02/07	*Team meeting*		
02/08	Mutation-based testing	Reading 1 and 2	Beta release (due 02/14)
02/09	*Project meeting*		
02/10	**In-class exercise (Code defenders)**	Canvas	
02/13	***Hack day***		
02/15	Reflection		Implementation and Documentation (due 02/21)
02/16	*Project meeting*		
02/17	**In-class exercise (Testing)**	Canvas	
02/20	***No class (holiday)***		
02/21	*Team meeting*		
02/22	Debugging	Reading	Peer review (due 02/28)
02/23	*Project meeting*		
02/24	**In-class exercise (Debugging)**	Canvas	
02/27	Program analysis		
02/28	*Team meeting*		
03/01	Fault localization		Final release (due 03/07)
03/02	*Project meeting*		
03/03	**In-class exercise (Fault localization)**	Canvas	
03/06	***Hack day***		
03/08	Advanced program analysis		Reflection (due 03/14)
03/09	*Project meeting*		
03/10	Optional in-class exercise	Canvas	

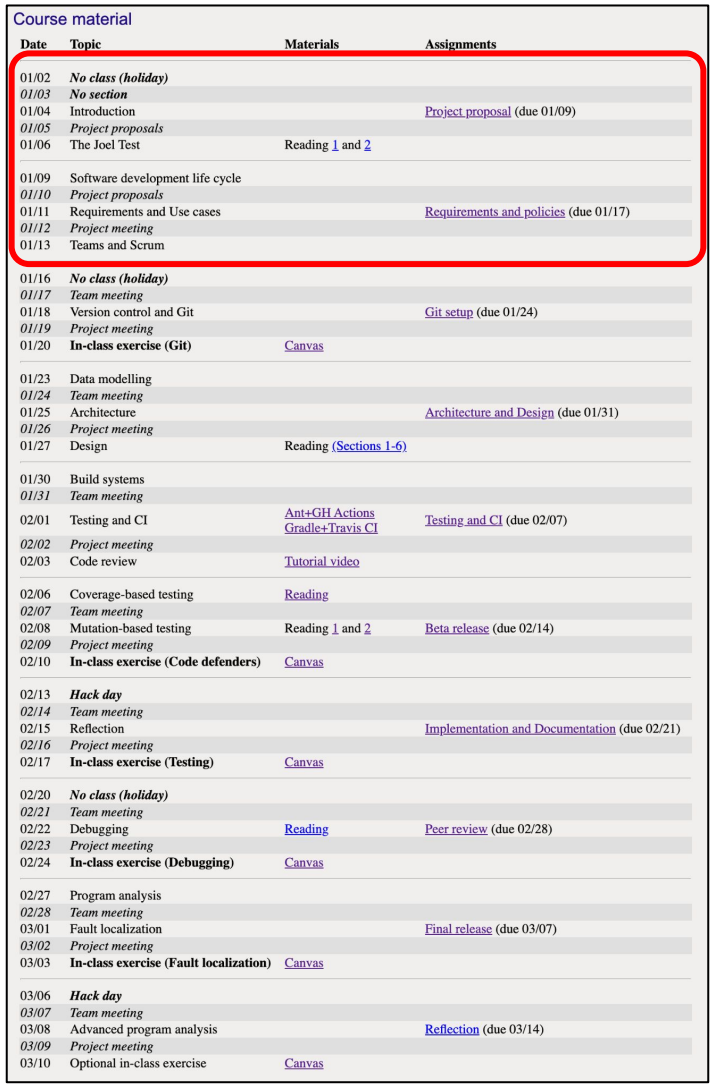
Grading

- 55%: Course project
- 20%: In-class exercises and individual assignments
- 15%: Mid-term exam
- 10%: Participation

Workload

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

Course overview: topics



Course material

Date	Topic	Materials	Assignments
01/02	***No class (holiday)***		
01/03	***No section***		
01/04	Introduction		Project proposal (due 01/09)
01/05	*Project proposals*		
01/06	The Joel Test	Reading 1 and 2	
01/09	Software development life cycle		
01/10	*Project proposals*		
01/11	Requirements and Use cases		Requirements and policies (due 01/17)
01/12	*Project meeting*		
01/13	Teams and Scrum		
01/16	***No class (holiday)***		
01/17	*Team meeting*		
01/18	Version control and Git		Git setup (due 01/24)
01/19	*Project meeting*		
01/20	**In-class exercise (Git)**	Canvas	
01/23	Data modelling		
01/24	*Team meeting*		
01/25	Architecture		Architecture and Design (due 01/31)
01/26	*Project meeting*		
01/27	Design	Reading (Sections 1-6)	
01/30	Build systems		
01/31	*Team meeting*		
02/01	Testing and CI	Ant+GH Actions Gradle+Travis CI	Testing and CI (due 02/07)
02/02	*Project meeting*		
02/03	Code review	Tutorial video	
02/06	Coverage-based testing	Reading	
02/07	*Team meeting*		
02/08	Mutation-based testing	Reading 1 and 2	Beta release (due 02/14)
02/09	*Project meeting*		
02/10	**In-class exercise (Code defenders)**	Canvas	
02/13	***Hack day***		
02/14	*Team meeting*		
02/15	Reflection		Implementation and Documentation (due 02/21)
02/16	*Project meeting*		
02/17	**In-class exercise (Testing)**	Canvas	
02/20	***No class (holiday)***		
02/21	*Team meeting*		
02/22	Debugging	Reading	Peer review (due 02/28)
02/23	*Project meeting*		
02/24	**In-class exercise (Debugging)**	Canvas	
02/27	Program analysis		
02/28	*Team meeting*		
03/01	Fault localization		Final release (due 03/07)
03/02	*Project meeting*		
03/03	**In-class exercise (Fault localization)**	Canvas	
03/06	***Hack day***		
03/07	*Team meeting*		
03/08	Advanced program analysis		Reflection (due 03/14)
03/09	*Project meeting*		
03/10	Optional in-class exercise	Canvas	

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

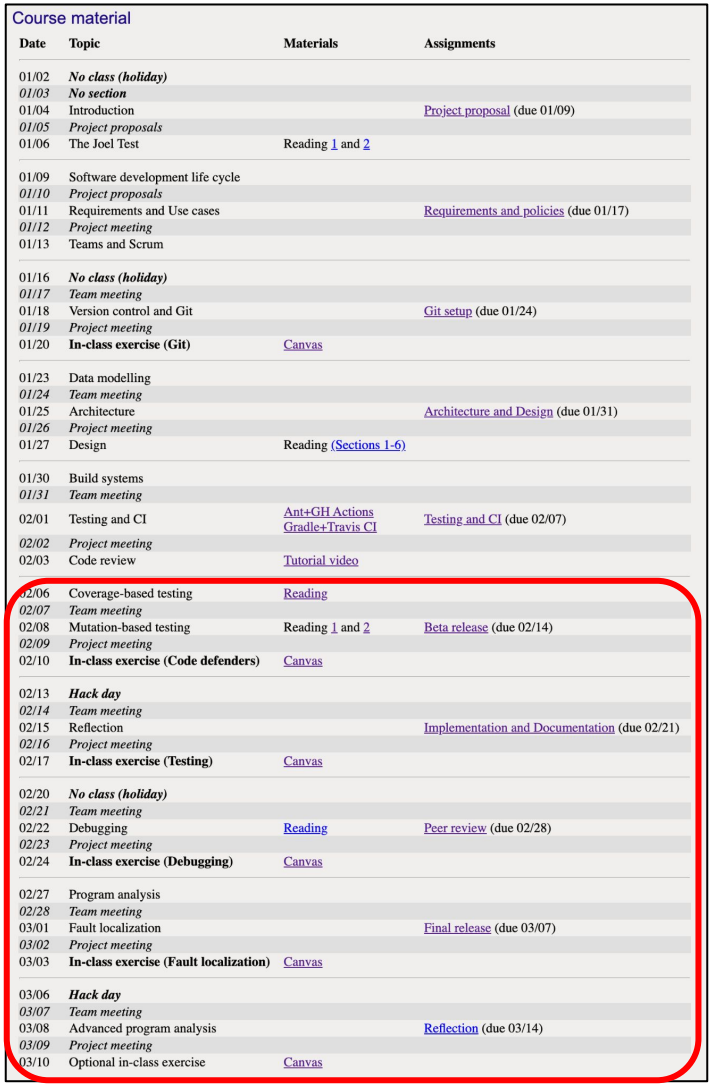
Course overview: topics

Course material

Date	Topic	Materials	Assignments
01/02	***No class (holiday)***		
01/03	***No section***		
01/04	Introduction		Project proposal (due 01/09)
01/05	*Project proposals*		
01/06	The Joel Test	Reading 1 and 2	
01/09	Software development life cycle		
01/10	*Project proposals*		
01/11	Requirements and Use cases		Requirements and policies (due 01/17)
01/12	*Project meeting*		
01/13	Teams and Scrum		
01/16	***No class (holiday)***		
01/17	*Team meeting*		
01/18	Version control and Git		Git setup (due 01/24)
01/19	*Project meeting*		
01/20	**In-class exercise (Git)**	Canvas	
01/23	Data modelling		
01/24	*Team meeting*		
01/25	Architecture		Architecture and Design (due 01/31)
01/26	*Project meeting*		
01/27	Design	Reading (Sections 1-6)	
01/30	Build systems		
01/31	*Team meeting*		
02/01	Testing and CI	Ant+GH Actions Gradle+Travis CI	Testing and CI (due 02/07)
02/02	*Project meeting*		
02/03	Code review	Tutorial video	
02/06	Coverage-based testing	Reading	
02/07	*Team meeting*		
02/08	Mutation-based testing	Reading 1 and 2	Beta release (due 02/14)
02/09	*Project meeting*		
02/10	**In-class exercise (Code defenders)**	Canvas	
02/13	***Hack day***		
02/14	*Team meeting*		
02/15	Reflection		Implementation and Documentation (due 02/21)
02/16	*Project meeting*		
02/17	**In-class exercise (Testing)**	Canvas	
02/20	***No class (holiday)***		
02/21	*Team meeting*		
02/22	Debugging	Reading	Peer review (due 02/28)
02/23	*Project meeting*		
02/24	**In-class exercise (Debugging)**	Canvas	
02/27	Program analysis		
02/28	*Team meeting*		
03/01	Fault localization		Final release (due 03/07)
03/02	*Project meeting*		
03/03	**In-class exercise (Fault localization)**	Canvas	
03/06	***Hack day***		
03/07	*Team meeting*		
03/08	Advanced program analysis		Reflection (due 03/14)
03/09	*Project meeting*		
03/10	Optional in-class exercise	Canvas	

- **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

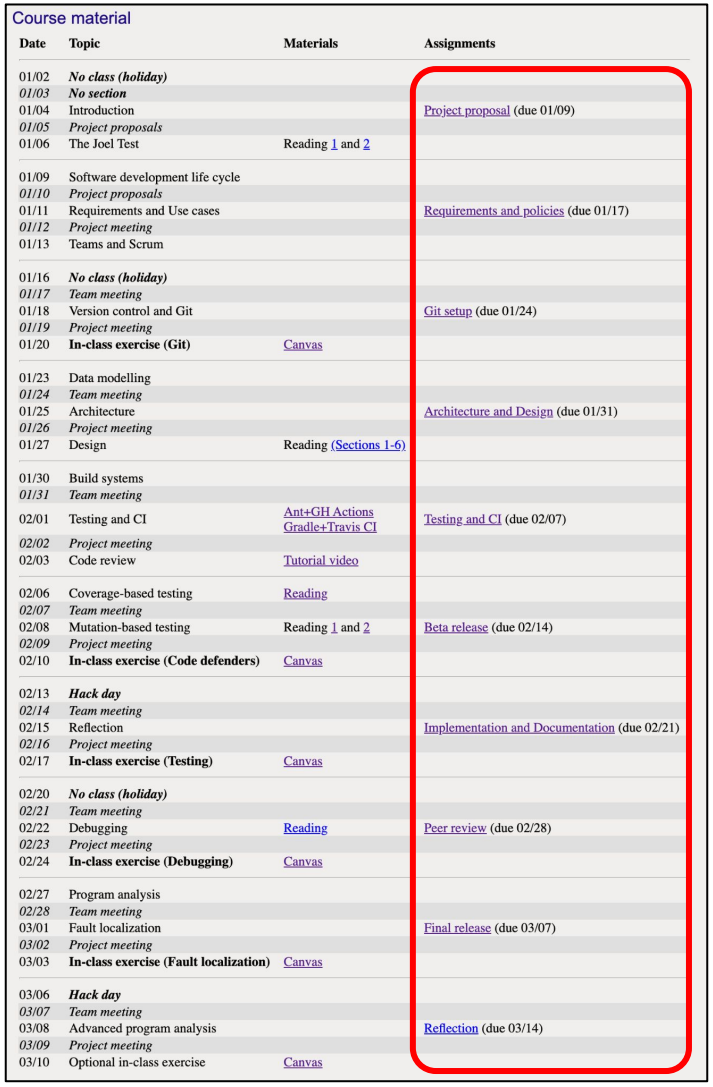


Course material

Date	Topic	Materials	Assignments
01/02	***No class (holiday)***		
01/03	***No section***		
01/04	Introduction		Project proposal (due 01/09)
01/05	*Project proposals*		
01/06	The Joel Test	Reading 1 and 2	
01/09	Software development life cycle		
01/10	*Project proposals*		
01/11	Requirements and Use cases		Requirements and policies (due 01/17)
01/12	*Project meeting*		
01/13	Teams and Scrum		
01/16	***No class (holiday)***		
01/17	*Team meeting*		
01/18	Version control and Git		Git setup (due 01/24)
01/19	*Project meeting*		
01/20	**In-class exercise (Git)**	Canvas	
01/23	Data modelling		
01/24	*Team meeting*		
01/25	Architecture		Architecture and Design (due 01/31)
01/26	*Project meeting*		
01/27	Design	Reading (Sections 1-6)	
01/30	Build systems		
01/31	*Team meeting*		
02/01	Testing and CI	Ant+GH Actions Gradle+Travis CI	Testing and CI (due 02/07)
02/02	*Project meeting*		
02/03	Code review	Tutorial video	
02/06	Coverage-based testing	Reading	
02/07	*Team meeting*		
02/08	Mutation-based testing	Reading 1 and 2	Beta release (due 02/14)
02/09	*Project meeting*		
02/10	**In-class exercise (Code defenders)**	Canvas	
02/13	***Hack day***		
02/14	*Team meeting*		
02/15	Reflection		Implementation and Documentation (due 02/21)
02/16	*Project meeting*		
02/17	**In-class exercise (Testing)**	Canvas	
02/20	***No class (holiday)***		
02/21	*Team meeting*		
02/22	Debugging	Reading	Peer review (due 02/28)
02/23	*Project meeting*		
02/24	**In-class exercise (Debugging)**	Canvas	
02/27	Program analysis		
02/28	*Team meeting*		
03/01	Fault localization		Final release (due 03/07)
03/02	*Project meeting*		
03/03	**In-class exercise (Fault localization)**	Canvas	
03/06	***Hack day***		
03/07	*Team meeting*		
03/08	Advanced program analysis		Reflection (due 03/14)
03/09	*Project meeting*		
03/10	Optional in-class exercise	Canvas	

- **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	Assignments
01/02	***No class (holiday)***		
01/03	***No section***		
01/04	Introduction		Project proposal (due 01/09)
01/05	*Project proposals*		
01/06	The Joel Test	Reading 1 and 2	
01/09	Software development life cycle		
01/10	*Project proposals*		
01/11	Requirements and Use cases		Requirements and policies (due 01/17)
01/12	*Project meeting*		
01/13	Teams and Scrum		
01/16	***No class (holiday)***		
01/17	*Team meeting*		
01/18	Version control and Git		Git setup (due 01/24)
01/19	*Project meeting*		
01/20	**In-class exercise (Git)**	Canvas	
01/23	Data modelling		
01/24	*Team meeting*		
01/25	Architecture		Architecture and Design (due 01/31)
01/26	*Project meeting*		
01/27	Design	Reading (Sections 1-6)	
01/30	Build systems		
01/31	*Team meeting*		
02/01	Testing and CI	Ant+GH Actions Gradle+Travis CI	Testing and CI (due 02/07)
02/02	*Project meeting*		
02/03	Code review	Tutorial video	
02/06	Coverage-based testing	Reading	
02/07	*Team meeting*		
02/08	Mutation-based testing	Reading 1 and 2	Beta release (due 02/14)
02/09	*Project meeting*		
02/10	**In-class exercise (Code defenders)**	Canvas	
02/13	***Hack day***		
02/14	*Team meeting*		
02/15	Reflection		Implementation and Documentation (due 02/21)
02/16	*Project meeting*		
02/17	**In-class exercise (Testing)**	Canvas	
02/20	***No class (holiday)***		
02/21	*Team meeting*		
02/22	Debugging	Reading	Peer review (due 02/28)
02/23	*Project meeting*		
02/24	**In-class exercise (Debugging)**	Canvas	
02/27	Program analysis		
02/28	*Team meeting*		
03/01	Fault localization		Final release (due 03/07)
03/02	*Project meeting*		
03/03	**In-class exercise (Fault localization)**	Canvas	
03/06	***Hack day***		
03/07	*Team meeting*		
03/08	Advanced program analysis		Reflection (due 03/14)
03/09	*Project meeting*		
03/10	Optional in-class exercise	Canvas	

- **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

Create a *new* program/app/feature that scratches your itch.

Example categories

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

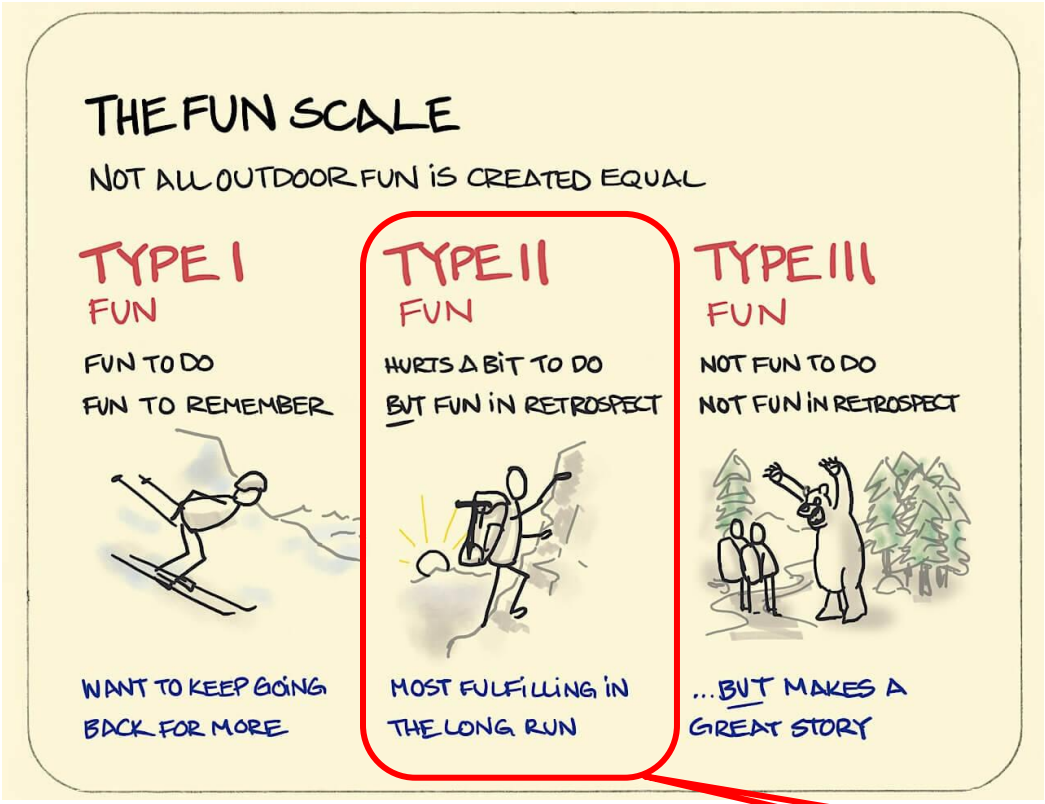
Example projects from 23au

IDE plugin to add print statements for logging.
It uses AI to determine what to print to explain the code.

Visualization of state representatives, over time, and how they voted on issues.

A Minecraft extension.

CSE 403: mostly type II fun



Sweet spot for teaching

Expectations

- Ability to program (in any programming language).
- Active participation in discussions.
- Teamwork and communication.
- Reflect on and improve your submissions.
- Go beyond adequate.

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
- Trade-offs, decisions, and justifications

CSE 403: challenges for staff

In-person vs. online education

2020: “Transition plan”

2022: “How does this work?”

Enrollment

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

Time

- Project duration: 9 weeks
- Lecture time: 50 minutes
- Quick turnaround times

CSE 403: challenges for students and staff

The Week-1 rush

Lecture time (12:30)

What's next?

- *Thu: Work on project proposal (pre-assigned groups)*
- Fri: The Joel Test (basic software engineering processes)