<section-header><section-header><section-header></section-header></section-header></section-header>	<ul> <li>Today</li> <li>The CSE 403 team</li> <li>Logistics and resources</li> <li>What is Software Engineering</li> <li>Course overview and expectations</li> </ul>
<ul> <li>The CSE 403 team</li> <li>Instructor</li> <li>René Just (rjust@cs.washington.edu)</li> <li>Office hours: After class and by appointment</li> <li>Deaching assistants/project managers</li> <li>Jesse Hu</li> <li>Ben Kushigian</li> <li>Edward Misback</li> <li>Reshabh Sharma</li> <li>Apollo Zhu</li> </ul>	Logistics: meetings • Lectures: M/W/F 12:30pm – 1:20pm (G10) • Team meetings: Tue 1:30pm – 2:20pm (ECE 125) • Project meetings: Thu 1:30pm – 2:20pm (G10)

This Thursday only: Work on project proposal with your assigned partner.

## Logistics: resources

- Course website: <u>https://homes.cs.washington.edu/~rjust/courses/CSE403</u> (cs.uw.edu/403)
- Submission of assignments via **Canvas**: <u>https://canvas.uw.edu</u>
- Discussions on Slack: <u>https://cse403-wi23.slack.com</u>

# Logistics: communication

## **Communication guidelines**

- We use Slack for all **non-sensitive** communication.
- See the <u>Slack guidelines</u> for this course.

### Resources

- The go-to page for this course is the <u>course web site</u>.
- 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?





# 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!

## Why is Software Engineering important?

## Software is eating the world!



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!
- $\circ$   $\,$  Software testing and debugging
- Maintenance and refactoring

## 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.

## The Role of Software Engineering in Practice



(Engineering workflow at Microsoft, Big Code summit 2019)

## The Role of Software Engineering in Practice



(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	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 proposale		
01/11	Requirements and Use cases		Requirements and policies (due 01/17)
01/12	Project mosting		the second s
01/13	Teams and Scrum		
01/16	No clarr (holidar)		
01/17	Them marting		
01/18	Varian control and Git		Git sature (due 01/24)
01/10	Pasiest meeting		Survey (are or any
01/20	In altern anomies (Cit)	Comment	
0020	means exercise (Gil)	S. HEYAS	
01/23	Data modelling		
01/24	Team meeting		
01/25	Architecture		Architecture and Design (dae 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	Coverson-based testing	Reading	
02007	Teom meeting	E22050010-6	
02/02	Materian based testing	Reading Land 2	Pata miana (dea 02/14)
02/08	Period meeting	Keasing Lana 2	Dent House (due our 14)
02/10	In-class exercise (Code defenders)	Canvas	
00/12	Rech des		
02/13	Hack bay		
00/15	Dedention .		Inclusion and Decomposition (dec 03/21)
02/15	President and		imprementation and polyumentation (due 02/21)
02/17	Troper meening	C	
02/17	in-cuss exercise (Testing)	CHINAS	
02/20	No class (holiday)		
0.021	Team meeting		
02/22	Deputzing	Research	Peer review (due 02/28)
02/23	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

٠

- 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



#### 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

0	a second a stat		
Course	e materiai		
Date	Topic	Materials	Assignments
01.02	No. down of a linear		
01/02	No cuts (nonady)		
07/05	No section		B 1
01/04	Introduction		Project proposal (due 01/09)
07/05	Project proposals		
01/05	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	Jeam meeting		
01/18	Version control and Gil		CitLachup (due 01/24)
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	Term meeting		
41777	Ranning	AntaGH Actions	
02/01	Testing and CI	Gradle+Travis CI	Testing and CI (due 02/07)
02/02	Project meeting		
02/03	Code review	Tutorial video	
02/05	Coverage-based testing	Reading	
02/07	Team meeting		
02/08	Mutation-based testing	Reading 1 and 2	Beta release (due 02/14)
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)
02/17	In-class exercise (Testing)	Canvas	
02/20	No class (holiday)		
02/21	Team meeting		
02/22	Debugging	Reading	Peer review (due 02/28)
00.04	In oher sparsies (Debugging)	Comme	
0024	in-case exercise (into agging)	Carra	
02/27	Program analysis		
02/28	Team meeting		
03/01	Fault localization		Final release (due 03/07)
02002	Desired meeting		
00/03	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
- 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

#### 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

Dete	Tests	Meteolale	1 selements
Date	Торіс	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 mosting		
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	
01/22	Data modalling		
01/25	Data moderning		
01/24	Ambitrature		Ambitrature and Desire (dee 01/21)
01/25	Paralant mention		Anomoved and LASSIN (due 01/31)
01/26	Project meeting		
0027	rvesign	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		
03007	Team meeting		
03/08	Advanced morenam analysis		Reflection (due 03/14)
03/09	Project meeting		
03/10	Ontional in class exercise	Conva	
	optional in cash catterine	School of the local school	

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

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 (dae 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	Teommetine		
01/25	Architecture		Architecture and Desirn (dan 01/31)
01/26	Project meeting		Contraction of the second (one of or ()
01/27	Design	Reading (Sections 1-6)	
01/30	Build systems		
01/31	Term meeting		
02/01	Testing and CI	Ant+GH Actions	Testing and CI (due 02/07)
		Gradle+Travis CI	
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 (holidar)		
02/21	Team meeting		
02/22	Debugging	Reading	Peer review (due 02/28)
02/28	Project meeting		
	In abore exercise (Debuosine)	Country	
02/24	metals exercise (Debugging)	S. MILLING.	
02/24	Program analysis	Laton	
02/24 02/27 02/27	Program analysis Toom meeting	Carra	
02/24 02/27 02/28 03/01	Program analysis Team meeting Fault localization	LINCOM	Pinal release (dae (1307)
02/24 02/27 02/28 03/01 03/02	Program analysis Team meeting Fault localization Project meeting	L.H.L.M.	Final release (due 0307)
02/24 02/27 02/28 03/01 03/02 03/03	Program analysis Team meeting Fault localization Project meeting In-chase secreting	Cierca	Pinal release (dae 03/07)
02/24 02/27 02/28 03/01 03/02 03/03	Program analysis Team meeting Pault localization Project resetting In-class exercise (Fault localization)	Canvas	<u>Pinal telease</u> (due 03/07)
02/24 02/27 02/28 03/01 03/02 03/03 03/03	Program analysis Program analysis Fount neering Foult localization Project reserting In-class secretise (Fault localization) Hack day	Canvas	Einal release (due 03/07)
02/24 02/27 02/28 03/01 03/02 03/03 03/06 03/06	Program analysis Program analysis Prom meeting Pault localization Project monting In-class exercise (Fault localization) Hack day Toom meeting	Canvas	Einal release (due 03/07)
02/24 02/27 02/28 03/01 03/02 03/03 03/06 03/06 03/07 03/08	Program anilysis Toom meeting Fault localization Project needing In-class exercise (Fault localization) Hack day Toom meeting Advanced program analysis	Canvas	Einal release (dae 03/07) Reflection (dae 03/14)
02/24 02/27 02/28 03/01 03/02 03/03 03/06 03/07 03/08 03/09	Program analysis Toom meeting Fault localization Project needing In-class exercise (Fault localization) Hack day Toom meeting Advanced program analysis Project meeting	Canvas	Einal release (due 03/07) Reflection (due 03/14)

Course overview: topics

Course material Date Topic

#### 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

	material		
Deste	Tests	Meteriale	A select month
Date	торіс	staterias	Assignments
01/02	No class (holiday)		
01/03	No section		
01/04	Introduction		Project proposal (due 01/09)
01/05	Project proposals		
01/05	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 (holidas)		
01/17	Team meeting		
01/18	Version control and Git		Git setue (due 01/24)
01/19	Project meeting		
01/20	In-class exercise (Git)	Canvas	
01/23	Data modalling		
01/24	Teammenting		
01/25	Architecture		Architecture and Desirn (due 01/31)
01/26	Project meeting		
01/27	Design	Reading (Sections 1-6)	
01/30	Build systems		
01151	160m meeting	AntaGU Actions	
02/01	Testing and CI	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		
02/13 02/14	Hack day Team meeting		
02/13 02/14 02/15	Hack day Team meeting Reflection		Implementation and Documentation (due 02/21)
02/13 02/14 02/15 02/16	Hack day Team meeting Reflection Project meeting		Implementation and Documentation (due 02/21)
02/13 02/14 02/15 02/16 02/17	Hack day Team meeting Reflection Project meeting In-class exercise (Testing)	Canvas	Implementation and Documentation (due 02/21)
02/13 02/14 02/15 02/16 02/17 02/20	Hack day Team meeting Reflection Project meeting In-class exercise (Testing) No class (holidar)	Canvas	Implementation and Documentation (due 02/21)
02/13 02/14 02/15 02/16 02/17 02/20 02/21	Hack day Toom mering Reflection Project meeting In-class exercise (Testing) No class (koliday) Toom meeting	Canvas	Implementation and Documentation (due 02/21)
02/13 02/14 02/15 02/16 02/17 02/20 02/21 02/22	Hack day Toom meeting Reflection Project meeting In-class exercise (Testing) No class (koliday) Toom meeting Debugging	Canvas	Implementation and Documentation (dwe 02/21) Peer review (due 02/28)
02/13 02/14 02/15 02/16 02/20 02/20 02/21 02/22 02/23	Hack day Toam mseering Reflection Project meeting In-class exercise (Testing) No class (holiday) Toam mseering Dobugging Project meeting	Canvas Reading	Implementation and Documentation (doe 02/21)
02/13 02/14 02/15 02/16 02/17 02/20 02/21 02/22 02/23 02/24	Hack day Taom mering Reflection Project meeting In-class exercise (Testing) No class (holiday) Toom mering Debagging Project meeting In-class exercise (Debugging)	Canvas Reading Canvas	Implementation and Documentation (due 02/21) Peer review (due 02/28)
02/13 02/14 02/15 02/16 02/17 02/20 02/21 02/22 02/23 02/24	Hack day Taom maring Reflection Project meeting In-class exercise (Testing) No class (holiday) Toom meeting Debugging Project meeting In-class exercise (Debugging) In-class exercise (Debugging) Program subvise	Canvas Reading Canvas	Implementation and Decumentation (she 0221) Percencyion (she 0228)
02/13 02/14 02/15 02/16 02/17 02/20 02/21 02/22 02/22 02/23 02/24 02/27 02/28	Hack dey Taom mering Reflection Project meeting In «base scoreting (Testing) In «base networks) Debugging Debugging Debugging In «base scoreting In «base scoreting (Debugging)	Canvas Reading Canvas	Implementation and Decumentation (due 0221) Pare mysion (due 0228)
02/13 02/14 02/15 02/16 02/17 02/20 02/21 02/22 02/23 02/24 02/27 02/28 03/01	Hack day Taom marring Taom marring Reflection Project meeting In-class exercise (Testing) No class (holday) Toom meeting Debugging Project meeting In-class exercise (Debugging) Project meeting Toom me	Canvas Reading Canvas	Implementation and Documentation (due 0221) Part project (due 0228) Final release (due 0307)
02/13 02/14 02/15 02/17 02/20 02/21 02/22 02/22 02/24 02/27 02/24 02/27 02/28 03/01 03/02	Hack day Taom mering Reflection Project meeting In-class exercise (Testing) Debugging Project meeting Debugging Project meeting In-class exercise (Debugging) Project meeting Toam meeting Fram meeting	Canvas Reading Canvas	Implementation and Documentation (dwe 0271) Part restor: (dwe 0273) Emil relate: (dwe 0307)
02/13 02/14 02/15 02/17 02/20 02/21 02/22 02/22 02/24 02/24 02/27 02/28 03/01 03/02 03/03	Hack dey Hack dey Reflection Project resord In class (heliday) Tom merring Dougging Dougging Dougging In-class secretice (Dolugging) Program snityiss Degram snityiss Degram snityiss Desrets concerning Project monting Declass secretice (Dalu localization)	Canvas Resoling Canvas	Implementation and Documentation (due 02/21) Percreption (due 02/28) Einst rations (due 03/07)
02/13 02/14 02/15 02/16 02/17 02/20 02/21 02/22 02/24 02/27 02/28 03/01 03/02 03/03 03/04	Hack day Hack day Team maring Team maring Team maring Team across (Testing) Team maring Debugging Debuggin	Canvas Resting Canvas	Implementation and Documentation (der 0021) Pour notes: (der 0020) Final mittens: (der 0037)
02/13 02/14 02/15 02/16 02/17 02/20 02/21 02/22 02/22 02/24 02/27 02/28 03/03 03/06 03/06 03/06	Hack day Team norting Reflection Team norting Programmer Nortice (Testing) Nortice (Testing) Nortice (Testing) Nortice (Testing) Nortice (Testing) Programmer Nortice (Testing) Programmer Nortice (Testing) Nortice (Testing) Testing (Testing) Nortice (Testing) Testing (Testing) Nortice (Testing) Testing (Testing) Nortice (Testing) Nor	Canvas Resslar Canvas	Intermetation and Documentation (dow 02/21) Percruties (dow 02/20) Pand relates (dow 03/07)
02/13 02/14 02/16 02/17 02/20 02/21 02/22 02/22 02/22 02/24 02/27 02/28 03/01 03/03 03/06 03/06 03/08	Hack day Taom marking Taom Taom Tao Taom Tao Tao Tao Tao Tao Tao Tao Tao Tao Tao	Canvas Resslins Canvas	Instementation and Decommunicat (der 0021) Part resider (der 0028) Pant relieur (der 0007) Reflecture (der 0007)
02/13 02/14 02/16 02/17 02/20 02/21 02/22 02/23 02/24 02/27 02/28 03/01 03/02 03/03 03/06 03/06 03/09	Hack day Taom marking Taom marking Taom marking Taom marking Taodase acreates Taom marking Taoma	Canvas Reading Canvas	Immemetation and Documentation (doe 0021) Parameters (doe 0220) Enalations: (doe 0307) Reflecting (doe 0304)
02/13 02/14 02/15 02/16 02/17 02/20 02/21 02/23 02/24 02/27 02/28 03/01 03/03 03/03 03/06 03/08 03/08 03/08	Hack day Team serving Team serving Canan serving Leanse ceretor Cetangu Des Leanse Cetangu Des Leanse Cetangu Des Leanse Des Leanse Cetangu Des Leanse Leanse Des Leanse D	Canzas Canzas Canzas	Internetation and Documentation (der 0271) Part review (der 0273) Part review (der 0273) Reduction (der 0307) Reflection (der 0314)

# • 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.

Date	торіс	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	

## Course overview: course project

- 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
  - $\circ$   $\;$  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



## 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
- Trade-offs, decisions, and justifications

# CSE 403: challenges for staff



# What's next?

- Thu: Work on project proposal (pre-assigned groups)
- Fri: The Joel Test (or why you really should take 403)