

Software Engineering

Questions & Learning Goals

- When is a program good enough to ship? Have you built what the customer wanted?
- You can write code. Can you build software?
- How to efficiently collaborate with teammates in a distributed setting?
- How do you get a patch accepted into an open-source project?
- What are the research topics in the Software Engineering area?

(This course is a significant redesign of previous ECE444 (2019) and heavily inspired by the CMU Software Engineering courses [15-313](#) and [15-214](#). Many thanks to Prof. [Michael Hilton](#), Prof. [Christian Kästner](#), and Prof. [Claire Le Goues](#) for generously sharing the course materials and providing valuable advices.)

Overview

Successful software projects require more than just technical expertise. Figuring out what the client wants, collaborating in a team, managing complexity, mitigating risks, staying on time and budget, and determining under various constraints when a product is good enough to be shipped are at least equally important topics that often have a significant human component. ECE444 explores these issues broadly covering the fundamentals of modern software engineering.

Assuming reasonably solid programming skills (including unit testing and code-level design), we will explore the following topics:

- **Process consideration for software development**

(How do avoid problems early? When and how much to design? When and how much to test? When and how to involve the customers? Agile methods...)

- **Requirements elicitation, documentation, and evaluation**

(How to figure out what the customer really wants? Who else has an interest? How can we measure success objectively? How can we reliably document expectations? ...)

- **Design for quality attributes**

(How can we design a system to be able to scale to millions of users? How can we design security into a system? ...)

- **Strategies for quality assurance**, including measurement, inspection, and static and dynamic analysis

(What quality assurance strategy is best for a given system? What can we automate and when should we keep humans in the loop? How much testing and what kind of testing should we do? What qualities are important to assure beyond functional correctness? Can we evaluate usability, scalability, reliability, performance? How can we statically guarantee the absence of certain security issues? ...)

- **Empirical methods in software engineering**

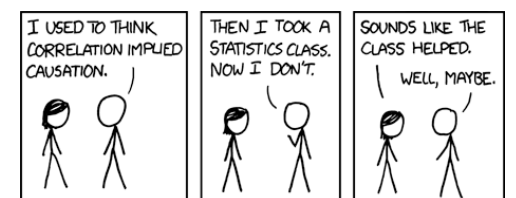
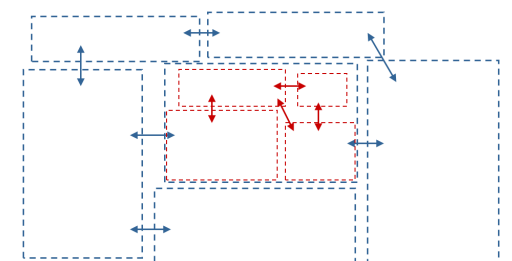
(How can we measure quality attributes such as performance, security, and reliability? How can we measure how users interact with the system? How can we know whether the difference matters? ...)

- **Time and team management**

(How to estimate the duration and costs of a project? How to monitor progress and risks to recognize issues early? How to coordinate developers in a team? How to form and develop teams? How to select and motivate team members? How to deal with team dynamics such as social loafing? ...)

- **Software Engineering Research**

(Research in Software Engineering studies how we build software that repairs itself, intelligently adapts to a changing world, and is trustworthy in a world full of dangerous situations and adversaries. It explores how to create a world where everyone can harness the power of programming and how to help software teams to build high quality software systems. In this course, we will link each of the topics with the state-of-the-art SE research projects.)



This course has a strong technical focus, and students will get experience with team management and modern software-engineering tools. The course puts students on a fast track toward project management positions.

"It's kind of like a root canal: you waited till the end, there are things you could have done beforehand. It's like preventative healthcare, but it's preventative software." -- [Margaret Hamilton](#)

Course Project Show Case [Fall 2020]

Project 1 (Web Application) - Chef's Co-Pilot

[Plate Up](#)

[Group 4's Chef Co-Pilot](#)

[Group 6's Chef Co-Pilot](#)

[ezcook](#)

Project 2 (Open Source Excursion)

Below is a list of merged PRs contributed by our students.

<https://github.com/bokeh/bokeh/pull/10757>

<https://github.com/bokeh/bokeh/pull/10743>

<https://github.com/ManimCommunity/manim/pull/817>

<https://github.com/ManimCommunity/manim/pull/869>

<https://github.com/RasaHQ/rasa/pull/7404>

<https://github.com/HospitalRun/hospitalrun-frontend/pull/2526>

<https://github.com/mydraft-cc/ui/pull/61>

https://github.com/dkkocab/WebSiteBot_PS5/pull/22

<https://github.com/Arcanorum/dungeonz/pull/82>


Logistics and People

Lectures: Mon/Wed/Fri 14:00-15:00 EST

Lab-1: Thur 12:00-15:00


Lab-2: Wed 09:00-12:00


Office hour: Fri 16:00-17:00 (Other time by appointment)

 Instructor: [Shurui Zhou](#)
shuruiz@ece.utoronto.ca




 TA: [Akshay Kawlay](#)
ak.kawlay@mail.utoronto.ca


 TA: [Yunhao Mao](#)
yunhao.mao@mail.utoronto.ca

 TA: [Karthik Mohan](#)
karthik.mohan@mail.utoronto.ca



 TA: [Keerthi Nelaturu](#)
keerthi.nelaturu@mail.utoronto.ca

 TA: [Jiten Saini](#)
jiten.saini@mail.utoronto.ca

 TA: [Haotian Zhao](#)
haotian.zhao@mail.utoronto.ca

Learning in a Global Pandemic

This course offers a place where students come together to discuss and debate ideas, learn about themselves and each other, and build community. Although we cannot come together in person this semester, the course team will work hard to support you and your learning, and do our best to cultivate a sense of community.

[Professor Brandon Bayne](#) of religious studies at the University of North Carolina at Chapel Hill produced a new syllabus for the summer 2020 edition of his course, addressing the pandemic directly. We've borrowed [his ideas](#), with gratitude to Professor Bayne:

- **Nobody signed up for this.**

- Not for the sickness, not for the social distancing, not for the sudden end of our collective lives together on campus

- Not for an online class, not for teaching remotely, not for learning from home, not for mastering new technologies, not for varied access to learning materials

- **The humane option is the best option.**

- We are going to prioritize supporting each other as humans



we are going to prioritize sharing resources and communicating clearly

- **We cannot just do the same thing online.**
 - Some assignments are no longer possible
 - Some expectations are no longer reasonable
 - Some objectives are no longer valuable
- **We will foster intellectual nourishment, social connection, and personal accommodation.**
 - Accessible asynchronous content for diverse access, time zones, and contexts
 - Optional synchronous discussion to learn together and combat isolation
- **We will remain flexible and adjust to the situation.**
 - Nobody knows where this is going and what we'll need to adapt
 - Everybody needs support and understanding in this unprecedented moment

Important Notes on Information Security Risk and Teaching Remotely from UofT. If you are a citizen of another country, and/or accessing your courses at the University of Toronto from a jurisdiction outside of Canada, please note that you may be subject to the laws of the country in which you are residing, or any country of which you have citizenship. The University of Toronto has a long-established commitment to freedom of expression, with this right enabled by an environment valuing respect, diversity, and inclusion. In your classes, you may be assigned readings, or discuss topics that are against the law in other jurisdictions. I encourage you to become familiar with any local laws that may apply to you and any potential impact on you if course content and information could be considered illegal, controversial, or politically sensitive. If you have any concerns about these issues, please contact your instructor directly to discuss with them.






Schedule

The following schedule describes the current planing status and the covered concepts. It is subject to change and will be updated as the semester progresses, especially to help focus on requested topics or support learning.

Lecture	Date	Topic	Reading assignments*	Important due*	
				Individual Participation	Group Project
1	9/11 F	Introduction		<div>Survey</div>	due <div>9/13</div> 11:59pm EST
2	9/14 M	Intro of Process & Team			
3	9/16 W	Requirement Engineering(RE) 1: Intro		<div>Proj1_Milestone0</div>	due <div>9/16</div> 11:59pm EST <div>Vote for idea</div> due <div>9/17</div> 11:59pm EST
4	9/18 F	Software Development Models	Goldstein, Harry (2005).	<div>PRA_1</div>	due <div>9/17</div> 11:59pm EST
 Lab		Git&GitHub			
5	9/21 M	Case Study & RE2: Elicitation		<div>Proj1_Milestone1</div>	: due <div>9/21</div> 11:59pm EST
6	9/23 W	RE 3: Doc, User Stories			
7	9/25 F	RE4: Story Mapping, Risk, Prototype & Measurements 1			
 Lab		Flask 1			
8	9/28 M	Measurements 2		<div>Proj1_Milestone2</div>	: due <div>9/27</div> 11:59pm EST

Individual Participation

Group Project

9	9/30 W	UML, OOP, Design Pattern 1	
10	10/2 F	Design Pattern 2 (SOLID).	
 Lab		Flask 2	
11	10/5 M	Design Pattern 3	<div>Proj1_Milestone3: due 10/5</div> 11:59pm EST
12	10/7 W	Architecture 1	
13	10/9 F	Inspections and Code Review & Architecture 2	<div>P1_M3 [PeerReview]: due 10/8</div> 11:59pm EST
 Lab		Docker Intro (2-week).	
	10/12 M	Thanksgiving Break	
14	10/14 W	Architecture 3	
15	10/16 F	Design Pattern 4	<div>Lab4-5 due 10/16</div> 11:59pm EST
 Lab		Docker Intro (2-week).	
16	10/19 M	Architecture 4 - Microservices	<div>Proj1_Milest.4: due 10/19</div> 11:59pm EST
17	10/21 W	Open Source 1	
18	10/23 F	[Guest Lecture] on Open Source by Richard Littauer	
 Lab		TDD (2-week).	
19	10/26 M	Mid-term presentation 1	
20	10/28 W	Mid-term presentation 2	
21	10/30 F	Mid-term presentation 3	<div>Lab6-7 due 10/30</div> 11:59pm EST
 Lab		TDD (2-week).	
22	11/2 M	DevOps	<div>Two-Week Report: Progress Update</div> : due 11/2 11:59pm EST
23	11/4 W	[Guest Lecture] founder of Gitcoin -- Kevin Owocki	
24	11/6 F	Quality Assurance 1	

Individual Participation

Group Project

🧪 Lab

25 11/16 [Quality Assurance 2](#)
M

26 11/18 [Quality Assurance 3](#)

Proj1_Milestone5: due 11/18

W 11:59pm EST

27 11/20 [Software Engineering for AI 1](#)
F

🧪 Lab

28 11/23 [Software Engineering for AI 2](#)
M

29 11/25 [Ethics](#)

Proj2_Milestone1: due 11/24

W 11:59pm EST

30 11/27 [Motivation&Teams](#)
F

🧪 Lab

31 11/30 [AI for SE](#)
M

32 12/2 [Software Engineering Research](#)
W

33 12/4 Project2 - Presentation 1
F

🧪 Lab

34 12/7 Project2 - Presentation 2

Proj2_Milestone1: due 12/19

M 11:59pm EST

35 12/9 Project2 - Presentation 3 + Summary
W

* For details, see assignment in Quercus.
PRA = Paper Reading Assignment. M = Monday, W = Wednesday, F = Friday. 11/9-11/13 Engineering Fall Study Break [\(link\)](#).

i Course Syllabus

The course uses [Quercus](#) for homework submission, grading, discussion, questions, announcements, and supplementary documents; slides will be posted here; [GitHub](#) is used to coordinate group work.

Expected Course Workload: Student workload should average to about 10 hours per week for ALL course-related activities [\(reference\)](#), including “contact” and/or “viewing” time, assignment/project completion, self-study, etc. In general, 3 hours/week will be spent in class and 7 hours on reading and project. Please feel free to give the course staff feedback on how much time the course is taking for you.

Teamwork: Teamwork is an essential part of this course. Most of the time spent working on this course will be spent working on the group project in teams of 3-5 students. There will be milestones for the project, and each milestone report has a component that is for the entire group and a component that is graded individually. Guidance on teamwork, reflection, and conflict resolution will be provided throughout the semester and are an essential component of the class. The team policy posted on Quercus applies and describes roles and teams and how to deal with conflicts and imbalances.

additional appointments.

Textbook: Various readings throughout the semester available online or through the library; we do not have a single text book but rather assemble readings from different sources.

☰ Assignments & Grading

Following the [FASE UofT Grading Policy](#), the evaluation of the course will be based on the following distribution: 60% a group project, 20% contributing to an open source project, 20% participation (reading quizzes).

- 📖 [Project 1 - 60%] A group project of **developing a web application** in teams of 3-5 students. There will be a series of milestones requiring group and individual reports (source code, documentation, commit history, etc). Some of the languages that will have to be used in conjunction with the project are Python, Flask, HTML, CSS, JavaScript, etc.
- 📖 [Project 2- 20%] A term project in which each team **contributes to an open source project** of their choice. This involves identifying an issue in the existing project, understanding the development process of that project and how to contribute, and actually making a contribution such as fixing a bug or adding a feature. Extra credit will be awarded if the contribution is merged into the project.
- 📖 (20%) Participation: Individual reading quizzes, Lab tasks.

Detailed Deliverables and Evaluation

- 📖 (60%) Web Application Development

Your mark will be determined by the quality of your design, source code, version control, documentation, how well you manage your project, and your mid-term and final presentations.

Milestone	Title	Marks	Due Date	Description
0	Meet your team & Proposing a project idea	2%	9/16	Submit 1-2 paragraphs (max 1/2 page) suggesting an online service worth having for group design and implementation in this class. Your submission will be evaluated based on: originality, understandability, feasibility.
1	Team workflow	3%	9/21	Complete and Sign Team Workflow document.
2	Requirements Elicitation	5%	9/27	Learn to conduct interviews with stakeholders and explore a problem space. Given what we discussed in class regarding how people conceptualize their experiences, design an interview script and conduct the interview with a relevant stakeholder for your class project.
3	Project Requirement Documentation & Peer Review	5%	10/5	Identify an appropriate scope for your system beginning with goals/objectives of the project. Synthesize requirements from various sources, including stakeholder interviews and other resources. Create user stories documenting the functional/quality requirements of the system. Create user interaction designs to clarify requirements. Learn to think critically and deliver constructive criticism.
4	Design and Coding (Midterm Report & Presentation)	20%	10/19	<ul style="list-style-type: none">• Start to structure your web application• Start to implement high priority user stories in your project• Following good coding practices• Mid-term presentation
5	Final Report & Architecture Report	25%	12/19	

- 📖 (20%) Open Source Excursion

Your high-level goal is to produce and submit a non-trivial modification or extension to an open-source project in a way that maximizes the chances that the project maintainers accept it. If you demonstrate to us that your change has been accepted and integrated into the project's code base, you will get up to 5(5%) bonus points. Your team will select an open source project, select a change to implement, actually contribute to the project, and present your insights to the class. You will individually reflect on your teamwork and open source experience.

1	Task Selection and Planning	10%	11/24
2	Project Report & Presentation	10%	12/19

Course Policies

Late work policy: Due to heavy reliance on teamwork in this course there are **no late days**. Exceptions to this policy will be made only in extraordinary circumstances, almost always involving a family or medical emergency---with your academic advisor or the Dean of Student Affairs requesting the exception on your behalf. Accommodations are possible if requested at least 3 days in advance. Please communicate also with your team about timing issues.

Notice of video recording and sharing (Download permissible; re-use prohibited) This course, including your participation, will be recorded on video and will be available to students in the course for viewing remotely and after each session. Course videos and materials belong to your instructor, the University, and/or other source depending on the specific facts of each situation, and are protected by copyright. In this course, you are permitted to download session videos and materials for your own academic use, but you should not copy, share, or use them for any other purpose without the explicit permission of the instructor. For questions about recording and use of videos in which you appear please contact your instructor.

Academic Integrity Policies and Collaboration: The usual policies apply, especially the [University Policy on Academic Integrity](#) and the [Code of Behaviour on Academic Matters](#) at UofT.

We expect that group members collaborate with one another, but that groups work independently from one another, not exchanging results with other groups. Within groups, we expect that you are honest about your contribution to the group's work. This implies not taking credit for others' work and not covering for team members that have not contributed to the team. Otherwise, our expectations regarding academic honesty and collaboration for group work are the same as for individual work, substituting elevated to the level of "group."

Here are some examples of behavior that are inappropriate:

- Copying or retyping, or referring to, files or parts of files (such as source code, written text, or unit tests) from another person or source (whether in final or draft form, regardless of the permissions set on the associated files) while producing your own. This is true even if your version includes minor modifications such as style or variable name changes or minor logic modifications.
- Getting help that you do not fully understand, and from someone whom you do not acknowledge on your solution.
- Writing, using, or submitting a program that attempts to alter or erase grading information or otherwise compromise security of course resources.
- Lying to course staff.
- Giving copies of work to others, or allowing someone else to copy or refer to your code or written assignment to produce their own, either in draft or final form. This includes making your work publicly available in a way that other students (current or future) can access your solutions, even if others' access is accidental or incidental to your goals. Beware the privacy settings on your open source accounts!
- Coaching others step-by-step without them understanding your help.

If any of your work contains any statement that was not written by you, you must put it in quotes and cite the source. If you are paraphrasing an idea you read elsewhere, you must acknowledge the source. Using existing material without proper citation is plagiarism, a form of cheating. If there is any question about whether the material is permitted, you must get permission in advance. We will be using automated systems to detect software plagiarism.

It is not considered cheating to clarify vague points in the assignments, lectures, lecture notes; to give help or receive help in using the computer systems, compilers, debuggers, profilers, or other facilities; or to discuss ideas at a very high level, without referring to or producing code.

Any violation of this policy is cheating. The *minimum* penalty for cheating (including plagiarism) will be a zero grade for the whole assignment. Cheating incidents will also be reported through University channels, with possible additional disciplinary action (see the above-linked University Policy on Academic Integrity).

If you have any question about how this policy applies in a particular situation, ask the instructors or TAs for clarification."

Note that the instructors respect honesty in these (and indeed most!) situations.

Inclusivity Statement: You belong [here](#). The University of Toronto commits to all students, faculty and staff that you can learn, work and create in a welcoming, respectful and inclusive environment. In this class, we embrace the broadest range of people and encourage their diverse perspectives. This team environment is how we will innovate and improve our collective academic success. You can read the evidence for this approach [here](#).

We expect each of us to take responsibility for the impact that our language, actions and interactions have on others. Engineering denounces discrimination, harassment and unwelcoming behaviour in all its forms. You have rights under the [Ontario Human Rights Code](#). If you experience or witness any form of harassment or discrimination, including but not limited to, acts of racism, sexism, Islamophobia, anti-Semitism, homophobia, transphobia, ableism and ageism, please tell someone so we can intervene. Engineering takes these reports extremely seriously. You can talk to anyone you feel comfortable approaching, including your professor or TA, an [academic advisor](#), our [Assistant Dean, Diversity, Inclusion and Professionalism](#), the [Engineering Equity Diversity & Inclusion Action Group](#), any staff member or a [U of T Equity Office](#).

You are not alone. [Here](#) you can find a list of clubs and groups that support people who identify in many diverse ways. Working together, we can all achieve our full potential.

Syllabus Statement on Accommodations: The University of Toronto supports accommodations for students with diverse learning needs, which may be associated with mental health conditions, learning disabilities, autism spectrum, ADHD, mobility impairments, functional/fine motor impairments, concussion or head injury, blindness and low vision, chronic health conditions, addictions, deafness and hearing loss, communication disorders and/or temporary disabilities, such as fractures and severe sprains, or recovery from an operation. If you have a learning need requiring an accommodation the University of Toronto recommends that students register as soon as possible with Accessibility Services at [here](#).

Phone: 416-978-8060

Email: accessibility.services@utoronto.ca

Mental Health Statement: As a university student, you may experience a range of health and/or mental health challenges that could result in significant barriers to achieving your personal and academic goals. Please note, the University of Toronto and the Faculty of Applied Science & Engineering offer a wide range of free and confidential services that could assist you during these times.

As a U of T Engineering student, you have an [Academic Advisor](#) (undergraduate students) or a [Graduate Administrator](#) (graduate students) who can support you by advising on personal matters that impact your academics. Other resources that you may find helpful are listed on the [U of T Engineering Mental Health & Wellness webpage](#), and a small selection are also included here:

- [Accessibility Services](#) & the [On-Location Advisor](#)
- [Graduate Engineering Council of Students' Mental Wellness Commission](#)
- [Health & Wellness](#) and the [On-Location Health & Wellness Engineering Counsellor](#)
- [Inclusion & Transition Advisor](#)
- [U of T Engineering Learning Strategist](#) and [Academic Success](#)
- [My Student Support Program \(MySSP\)](#).
- [Registrar's Office](#)
- [SKULE Mental Wellness](#)
- [Scholarships & Financial Aid Office & Advisor](#)

If you find yourself feeling distressed and in need of more immediate support resources, consider reaching out to the counsellors at [My Student Support Program \(MySSP\)](#) or visiting the [Feeling Distressed webpage](#).