CSE 403
Software Engineering
Spring 2023

#1: Course Introduction
Today

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

Instructor

- Nigini Oliveira (nigini @ cs)
- Office hours: After class and by appointment

Teaching assistants / project managers

- Reshabh K Sharma
- Vinay Reddy Varadha Pally
- Sahil Verma
- Mingyuan Zhong
- Apollo Zhu

Email us at: cse403-staff @ cs (Priority response!)
Today

- The teaching team
- **Logistics and resources**
- What is Software Engineering
- Course overview and expectations
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)
Logistics: resources

- **Course website:**
  [https://nigini.github.io/SWEng/offers/CSE403-SP23/](https://nigini.github.io/SWEng/offers/CSE403-SP23/) ([cs.uw.edu/403](cs.uw.edu/403))

- Submission of assignments via **Canvas:**
  [https://canvas.uw.edu](https://canvas.uw.edu) (course [1633262](1633262))

- Discussions on **Slack:**
  [https://cse403-sp23.slack.com](https://cse403-sp23.slack.com)
Logistics: communication

Communication guidelines
● We use Slack for all non-sensitive communication.
● See the Slack guidelines for this course.

Resources
● All relevant information is on the website, or linked from it.
● Canvas for assignments and non-public materials.
● And, remember: The Calendar page is your friend!
Today

- The teaching team
- Logistics and resources
- **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 are part of SW-E and much more!
What is Software Engineering?

“The engineering discipline (hence, uses science to improve applicability and efficiency) that is concerned with all aspects of software production.” — Ian Sommerville

- i.e.: specifying, designing, developing, analyzing, deploying, and maintaining a software system.
What is Software Engineering?

“An engineering discipline (hence, uses science to improve applicability and efficiency) that is concerned with all aspects of software production.” — Ian Sommerville

- i.e.: 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!

Tesla in fatal crash with firetruck was using automated driving system

Nation state hackers exploited years-old bug to breach a US federal agency

Boeing acknowledges flaws in its 787 cockpit simulator software

Security experts warn Android users as 'dangerous' new bug leaves users vulnerable

Facebook Patches Access Token Leak

Users should change their passwords to mitigate threats posed by the accidental leak of perhaps millions of account identity details.
SOFTWARE ENGINEER

What my friends think I do

What society thinks I do

What my mother thinks I do

What other engineers think I do

What I think I do

What I really do

Source: https://www.deviantart.com/rydain/art/What-I-Do-Software-Engineer-285663308
1.03. "Approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy or harm the environment. The ultimate effect of the work should be to the public good." – Joint ACM & IEEE professional code of ethics

Source: https://ethics.acm.org/code-of-ethics/software-engineering-code/
1.03. "Approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy or harm the environment. The ultimate effect of the work should be to the public good." – Joint ACM & IEEE professional code of ethics
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

Intro-level courses focus on the inner loop.

CSE 403 largely focuses on the outer loop.
Today

- The teaching team
- Logistics and resources
- What is Software Engineering
- Course overview and expectations
Course overview: 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

**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

|                | ! | ! | ! | ! | ! | ! | ! |

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

- **Software processes, requirements, and specification**
  - Different software development processes.
  - Precise writing (requirements and specifications).
- **Software development practice**
  - 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 it all in a group project.
W1: Setup & Tuning-in

<table>
<thead>
<tr>
<th>Date</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>03/27</td>
<td>L: Intro</td>
</tr>
<tr>
<td>03/28</td>
<td>T: Peers meetup</td>
</tr>
<tr>
<td>03/29</td>
<td>L: Projects</td>
</tr>
<tr>
<td>03/30</td>
<td>P: Proposals</td>
</tr>
<tr>
<td>03/31</td>
<td>L: Joel-Test</td>
</tr>
</tbody>
</table>
W2: Engineering 101

<table>
<thead>
<tr>
<th>Date</th>
<th>Activity</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/03</td>
<td>L: Dev. Cycle</td>
<td>DUE: PP.1.1!!</td>
</tr>
<tr>
<td>04/04</td>
<td>T: Proposals</td>
<td>DUE: PP.1.2!!</td>
</tr>
<tr>
<td>04/05</td>
<td>L: Requirements</td>
<td>Project Requirements (PR)</td>
</tr>
<tr>
<td>04/06</td>
<td>P: Requirements</td>
<td></td>
</tr>
<tr>
<td>04/07</td>
<td>L: Use-Cases</td>
<td></td>
</tr>
</tbody>
</table>
**W3: Tooling & Teaming**

<table>
<thead>
<tr>
<th>Date</th>
<th>L:</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/10</td>
<td>SCRUM</td>
</tr>
<tr>
<td>04/11</td>
<td>T:</td>
</tr>
<tr>
<td>04/12</td>
<td>L: Version Control</td>
</tr>
<tr>
<td>04/13</td>
<td>P:</td>
</tr>
<tr>
<td>04/14</td>
<td>LX: GIT</td>
</tr>
</tbody>
</table>

**DUE:** PR!!!

GitHub Project Setup (GPS)
W4: Engineering 201

WEEK 4
04/17  L: Data modeling
04/18  T:          DUE: GPS!!!
04/19  L: Architecture
04/20  P:          Design & Architecture (DnA)
04/21  L: Design
W5: Tooling reloaded

WEEK 5
04/24   L: Build Systems
04/25   T:
04/26   L: Testing  DUE: DnA!!!
04/27   P:
04/28   L: CI/CD   Testing & CI/CD (TCC)
W6: Testing, testing, and testing

WEEK 6
05/01  L: Test Coverage
05/02  T:                     DUE: TCC!!!
05/03  L: Mutation Testing   Alpha Release (R1)
05/04  P:                      
05/05  LX: Code Defenders   

W7: Hack & Reflect (have a talked about testing?)

<table>
<thead>
<tr>
<th>Date</th>
<th>Activity</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>05/08</td>
<td>L: Hack Day</td>
<td></td>
</tr>
<tr>
<td>05/09</td>
<td>T:</td>
<td>DUE: R1!!!</td>
</tr>
<tr>
<td>05/10</td>
<td>L: Course Reflection</td>
<td>Beta Release (R2)</td>
</tr>
<tr>
<td>05/11</td>
<td>P:</td>
<td></td>
</tr>
<tr>
<td>05/12</td>
<td>LX: Testing</td>
<td></td>
</tr>
</tbody>
</table>
## WEEK 8

<table>
<thead>
<tr>
<th>Date</th>
<th>Monday (L)</th>
<th>Tuesday (T)</th>
<th>Friday (P)</th>
<th>Thursday (L)</th>
<th>DUE</th>
<th>Release Peer-Review (RPR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>05/15</td>
<td>Code Review</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05/16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R2!!!</td>
</tr>
<tr>
<td>05/17</td>
<td>Debugging</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05/18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Release Peer-Review (RPR)</td>
</tr>
<tr>
<td>05/19</td>
<td>Debugging</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**W9: Advanced Techniques #2**

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>05/22</td>
<td>L: Hack Day</td>
<td></td>
</tr>
<tr>
<td>05/23</td>
<td>T:</td>
<td>RPR!!!</td>
</tr>
<tr>
<td>05/24</td>
<td>L: Fault Location</td>
<td>Final_Release (R3)</td>
</tr>
<tr>
<td>05/25</td>
<td>P:</td>
<td></td>
</tr>
<tr>
<td>05/26</td>
<td>LX: Fault Location</td>
<td></td>
</tr>
</tbody>
</table>
W10: Advanced Techniques #3

**WEEK 10**

05/29  

H: MEM-DAY

05/30  

T:       

DUE: **R3!!!**

05/31  

L: Program Analysis

Individual Reflexion (IR)

06/01  

P:       

06/02  

LX: PA (extra-cred)
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

Teaching
● 90 students
● ~ 18 projects
● 3 lectures to prep / week

Pace
● Grade & Feedback in two days
● Every week

Uncertainty
● No simple check-box grading
● Trade-offs, decisions, and justifications
CSE 403: challenges for students and staff

The Week-1 rush

Lecture time (12:30)
What’s next?

WEEK 1
03/27  L: Intro
03/28  T: Peers meetup
03/29  L: Projects
03/30  P: Proposals
03/31  L: Joel-Test
What’s next?

<table>
<thead>
<tr>
<th>WEEK 1</th>
<th>03/27</th>
<th>L: Intro</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>03/28</td>
<td>T: Peers meetup</td>
</tr>
<tr>
<td></td>
<td>03/29</td>
<td>L: Projects</td>
</tr>
<tr>
<td></td>
<td>03/30</td>
<td>P: Proposals</td>
</tr>
<tr>
<td></td>
<td>03/31</td>
<td>L: Joel-Test</td>
</tr>
</tbody>
</table>

Question, please!