CS 341: Software Engineering I

Fall 2022

Instructor:	Michael P. Rogers
Office:	Halsey 214
Office Hours:	M-F, 2-3PM, Other hours by appointment
Email:	rogersm@uwosh.edu
Phone:	920-424-1388 / 309-825-6454
Class Times:	TuTh 11:30-1:00 PM (Tu: Halsey 101C; Th: Halsey 309)
Credits:	3

Prerequisites: CS 271 (data structures) with a grade of C or better, or junior-level standing and CS 262 (OOD&P II) with a grade of C or better.

Description: This course will provide an in-depth study and analysis of at least one large scale software system. Students will analyze, design, and partially implement an extensive software project. Case studies will address major system concerns such as specification, classification, inter-relationships, validation, and evaluation. Other topics include the use of UML, prototyping, data flow diagrams and CASE tools.

Course Website: if it happens in this course, it will be posted on UWO's Canvas site. Set up notifications to be alerted when announcements are posted, new assignments, quizzes, or notes are posted and graded, etc.

Textbooks:
Required (*'d items are online only)

1. Fitzpatrick, Brian W. and Ben Collins-Sussman. *Debugging Teams: Better Productivity through Collaboration*, 1st Edition, O'Reilly Media.
2. Chacon, Scott and Ben Straub. Pro Git. Apress, 2014. *
3. Winters, Titus., Tom Manshreck, Hyrum Wright. Software Engineering at Google: Lessons Learned From Programming Over Time. Oreilly, 2020.*

Recommended:

1. Halvorsen, Hans-Petter. Software Development: A Practical Approach!
2. Mountain Goat Software
3. launchschool.com/books/agile_planning/
4. Richards, Mark, Neal Ford. *Fundamentals of Software Architecture*. O'Reilly Media, 2018.
5. Winters, Titus, Tom Manshreck, Hyrum Wright. *Software Engineering at Google*. O'Reilly Media, 2020.
6. Seemann, Mark. *Code That Fits in Your Head: Heuristics for Software Engineering*. Addison-Wesley, 2021.
7. Van Der Hoek, Andre. *Software Design Decoded: 66 Ways Experts Think*. MIT Press, 2016.

System Accounts:

1. GitHub
2. bit.io
3. Google

Course Outcomes :

Upon successful completion of the course, students will be able to:

1. Describe the concepts and principles that guide software engineering practice.
2. Explain the relationship between systems engineering and software engineering.
3. Define the requirements engineering process and associated activities for both computer based and web-based systems.
4. Define the analysis modeling process and associated activities for both computer-based and web-based systems.
5. Define the design engineering process and associated activities for both computer-based and web-based systems.
6. Describe testing strategies and techniques for computer-based and web-based systems.
7. Describe architectural design, styles, and patterns of computer-based and web-based systems.
8. Define the guiding principles of good user interface design.
9. Describe the use of component-level design in software engineering.
10. Describe the project management concepts applied to software engineering projects.
11. Apply software engineering principles and methods to produce a solution to a significant problem.
12. Select and apply appropriate analysis modeling practices and tools to a significant problem.
13. Select and apply appropriate requirements capture techniques and tools to a significant problem.
14. Select and apply appropriate design engineering techniques to a significant problem.
15. Design and implement an effective test strategy and plan.
16. Apply good user interface design principles to the development of a solution to a significant problem.
17. Identify appropriate situations for use of different software architectures.
18. Identify appropriate situations for use of components-based design.
19. Establish effective communication plans with clients and co-developers.
20. Conduct effective presentations for clients.
21. Conduct effective design reviews with co-developers.
22. Develop and adhere to project plans and schedules.
23. Seek timely assistance from others when individual efforts to resolve problems have been unsuccessful.
24. Establish an atmosphere of trust and mutual commitment with clients.

Grading Criteria:

Category	**%**
Exams (2)	30
Labs	20
Readings / Class Participation	10
Team Project	40

Grade Scale:

%	≥ 92	90-92	88-90	82-88	80-82	78-80	72-78	70-72	68-70	62-68	60-62	< 60
Letter	A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F

Major Due Dates:

Event	Date
Exam 1	Thursday, 10/13/22
Sprint 1	Tuesday, 10/18/22
Sprint 2	Tuesday, 11/8/22
Exam 2	Tuesday, 11/22/22
Sprint 3	Tuesday, 11/29/22
Final Presentation	Thursday, 12/15/22

Late Policies: Douglas Adams once said, "I love deadlines: I love the whooshing sound they make when they fly by". Hopefully you'll agree with the first part, and not the second, but regardless the policies are as follows:

1. Exams- If you are unable to take an exam at the prescribed time a) let me know in advance, via email, and b) provide justification (a note from medical professional who treated you, or a representative from the Dean of Students Office).
2. Labs, Reading Quizzes - late submissions will not be accepted (plan accordingly!)
3. Project milestones - see the Team Project section, below

Team Project: In this course, you will do more than just study software engineering concepts. You will put those concepts to work. In teams of 4, you will employ best agile practices to gather requirements, design, implement, and test a significant piece of software. The goal here is to, as best we can, simulate what happens in industry when teams of developers work on a product. The software is the dessert, the main course is the **process**, methodically, deliberately going through an agile development process.

You will be graded on a series of milestones and reports, and generally speaking they must be turned in by the due date. However, if you are falling behind, notify your instructor ahead of time, with an explanation of the issues and a recovery plan (fallback due date), for possible clemency. Notification after the due date will result in a 0.

Team Project Practices: All code must be stored in GitHub using appropriate branching strategies. Before any code is committed, to provide as much verisimilitude as possible, it must go through code review by another developer (or more!) (to check for quality and adherence to coding standards). All commit messages must make clear who a) wrote the code, b) who did the code review. This means that you cannot wait until the last moment to do your work, you will need to coordinate with team members.

Absences: It has been scientifically proven that the most significant factor for predicting student success is attendance (although whether this is truly causation or merely correlation is another question). You should attend each and every class as scheduled, and notify the instructor ahead of time if you will be absent. Attendance may be taken and factored into the Participation category.

Academic Integrity: The purpose of this course is to teach you in particular about software engineering, and more generally how to think, how to *problem solve*. For that reason, the work that you turn in must be your own. You may have *general* conversations with students to clarify the nature of an assignment, and you can ask for help with debugging, but that second-set-of-eyes-student should not be looking at their code while they assist you. Sometimes students are unaware of whether or not they have committed plagiarism, so here are some tips:

1. if your problem solving begins with ctrl-C and ends with ctrl-V, you have committed plagiarism.
2. if your problem solving starts at Google and ends at Chegg, you have committed plagiarism.
3. if your problem-solving involves surreptitiously glancing at the exam of the student next to you and doing a virtual copy-and-paste, you have committed plagiarism.
4. if that tiny voice inside your head, the same one that inconveniently shuts up entirely while you are working on a software engineering assignment, starts making "ahem" noises, you have committed plagiarism.

Let us consider the pros and cons of committing plagiarism.

Pros:

1. You have completed the assignment.

Cons:

1. You will have missed that exhilarating, ego-boosting, delicious "aha!" moment that everyone experiences when they have, on their own, solved a difficult puzzle.
2. You will be caught, receive a 0 on the assignment/exam, and may face disciplinary action in front of a bevy of grim-faced administrators who you do *not* want to meet.

So what do you do when you can't solve a problem? The answer is simple, ask your instructor for help. We will use the time-tested Socratic method, asking questions that will lead you to the correct answer. Failing that, just write "I have no idea" in bold letters in the comment block at the top of the assignment, turn it in, and you will gain *some* points (more than 0).

For more detailed information on what constitutes academic misconduct, please see the discussion of UWS Chapter 14, Student Academic Disciplinary Procedures.

Accessibility: Your instructor is committed to ensuring a fair playing field. If you have a disability and need assistance (e.g., a note taker, certain seating, extra time to take tests, adaptive technology, etc.), please register with the Accessibility Center, and we will accommodate your needs.

Non-discrimination and Anti-harassment: Your instructor is committed to maintaining a harassment-free, welcoming classroom, and will not tolerate discrimination on the basis of race, religion, creed, color, gender, identity/expression, ancestry, national origin, age, marital status, relationship to other employees, sexual orientation, disability, veteran's status, membership in the military, arrest/conviction record, political affiliation, or any other protected status.

HATE HAS NO HOME HERE.

Class Participation and Feedback: Your instructor relishes class participation and constructive feedback. If you are lost, please, please ask, during class. If you have been working in industry and have first-hand experience with what we are talking about, please chip in: everybody does it differently and it is interesting to get new perspectives.

Consumer Information: Students are advised to see the following URL for disclosures about essential consumer protection items required by the Students Right to Know Act of 1990: https://uwosh.edu/financialaid/consumer-information/

Syllabus Changes: if any substantive changes are made in the course syllabus, such as changes in schedule or assignments, notification will be provided in a timely manner and a revised syllabus made available.