CS 341: Software Engineering I
Fall 2023

Instructor: Michael P. Rogers
Office: Halsey 214
Office Hours: MW 3-4 PM, TR 2-3 PM, F 3-3:30 PM, or by appointment
Email: rogersm@uwosh.edu
Phone/Text: 309-825-6454

Class Times: TuTh 11:30-1:00 PM (Tu: Halsey 101C; Th: Halsey 208)
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:

1. Fitzpatrick, Brian W. and Ben Collins-Sussman. Debugging Teams: Better Productivity through Collaboration [Free, online]
2. Goldman, Matt. .NET MAUI in Action. Manning, 2023.
3. Chacon, Scott and Ben Straub. Pro Git. Apress, 2014. [Free, online]

References (good background information):

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. CockroachLabs
3. Google

Course Outcomes :
Upon successful completion of the course, students will be able to: Describe the concepts and principles that guide software engineering practice.

1. Describe the phases of software development - requirements gathering, analysis, design, implementation, testing, maintenance
2. Describe common software architecture paradigms, and be able to identify when each is appropriate
3. Define the guiding principles of good user interface design.
4. Describe the project management concepts applied to software engineering projects.
5. Apply software engineering principles and modern software engineering practices to produce a solution to a significant problem
6. Establish effective communication plans with clients and fellow-developers.
7. Conduct effective presentations for clients.
8. Participate in effective code reviews with fellow-developers.
9. Develop and adhere to project plans and schedules.
10. Establish an atmosphere of trust and mutual commitment with clients.

Grading Criteria:

Category	%
Quizzes	35
Labs	20
Readings/Attendance/Class Participation	5
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
Sprint 1	Tuesday, 10/17/23
Sprint 2	Tuesday, 11/7/23
Sprint 3	Tuesday, 11/28/23
Final Presentation	Thursday, 12/14/23

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. Quizzes - If you are unable to take a quiz 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 - late submissions will not be accepted (plan accordingly!)
3. Project milestones - see the Team Project section, below

Quizzes: There will be short, weekly quizzes, each covering the material discussed (or assigned as reading/viewing) during the previous week. Quizzes will take place on Tuesdays.

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: You may discuss problems with other students, but the work you turn in for labs and your project, and answers to quizzes, will be your own. For code snippets that you use for your team project, the source of those must be clearly labeled in the comments. Failure to adhere to these guidelines will result in a 0 for the lab/project/quiz in question. For detailed information on what constitutes academic misconduct, please see the discussion of UWS Chapter 14, Student Academic Disciplinary Procedures.

ChatGPT Policies: ChatGPT is good at explaining (some) CS topics, and, unlike your instructor, is available 24x7. So if you are unsure about a topic discussed in class, consider asking ChatGPT -- just remember that ChatGPT does not always get things right, so look at its answers with a skeptical eye. Also, to help with studying, consider using ChatGPT to generate practice problems (with solutions). What you may *not* use ChatGPT for is to complete your assignments: that is plagiarism (see above).

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.

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.