Course ID and Title: CSCI 577a – Software Engineering
Units: 4
Term, Day, and Time: Spring 2025 — Mon 6:00 – 9:20 PM
Location: OHE 132 + DEN

Co-Instructor: Mahdi Eslamimehr, PhD, MBA
Office: TBD
Office Hours: TBD
USC Email: eslamime

Co-Instructor: Jae Young Bang, PhD
Office: TBD
Office Hours: TBD
USC Email: jaeyounb

Teaching Assistant: TBD
Office: TBD
Office Hours: TBD
USC Email: N/A

Course Description

This is the first of two courses in the Software Engineering sequence, which serves as the core of the Master of Science in Computer Science with a specialization in Software Engineering. CSCI-577a covers the complete software development lifecycle, encompassing software planning, processes, requirements, architecture, product development, integration, testing, and maintenance.

The course is divided into two complementary parts, providing a comprehensive view of software engineering. Dr. Eslamimehr will focus on the management aspects of software engineering, incorporating real-world industry practices, while Dr. Bang will emphasize theoretical foundations with a focus on software architecture design.

Students will work in teams and are expected to apply quality management techniques to their projects. Each team will develop an operational MVP, which includes requirements specifications, architectures, prototypes, life cycle plans, and an integrating rationale for the proposed capabilities.

Additionally, student teams will be assigned cutting-edge research papers in software engineering. They are expected to critically review and evaluate these papers, demonstrating the analytical skills required of researchers and industry professionals when assessing new research.

Lecture Topics (Tentative)

Dr. Mahdi Eslamimehr: Software Engineering Management and Industry Practices

- What causes the software to fail?
    - History of software failures
    - Enterprises vs. startups
- Agile: state-of-industry
- Evaluation metrics & management
- Cloud nativeness
- AI is a commodity now

Dr. Jae Young Bang: Software Engineering Theory

- Introduction: “What is Software Engineering?”
- Software engineering philosophy
    - Software process models
    - Value-based software engineering
    - Affordability-based software engineering
- Incremental commitment spiral model
    - Exploration
    - Valuation
    - Foundations
    - Development
    - Operation
- Agile
    - Concepts
    - Quality assurance
    - Anti-patterns
    - Verification and validation
    - Scaling agile
- Analysis and planning
    - Risk analysis
    - Feasibility analysis
    - Software lifecycle plan
    - Cost estimation
    - Technical debt
- Software architecture design
    - “CSCI-578 in 90 minutes”
    - Architecture-based software engineering
- Prototyping UI/UX

- Reviews and testing
    - Evidence-based reviews
    - Software metrics & measurements
    - Quality management peer review
    - Software testing
- DevOps
    - Software version control
    - Continuous integration
    - Continuous testing
    - Continuous deployment
- Software maintenance
- Software engineering best practices
    - The “laws” of software engineering
- Software ethics
- Project management
- Future of software engineering

Student Evaluation Plan (Tentative)

- Exams (50%)
    - Exam #1 (25%)
    - Exam #2 (25%)
- Team projects(50%)
    - Team software development project (40%)
        - Delivery of minimum viable product (MVP)
    - Team presentations (10%)
        - Software engineering research paper presentations
        - Student teams will be provided with a list of recent software engineering papers and be assigned to one of those to present.

Grading Scale

Each exam, assignment, and project will be scaled appropriately and applied to the final letter grade. This means that you will not be able to project your letter grade based on your exam and assignment scores.

Assignment Submission Policy

Late work will be accepted with a penalty. The amount of points lost is calculated as follows:
`Points lost =` d^2, where `d` is the number of days late. Therefore, after being two days late, you lose four points; nine days late results in a loss of 81 points; and 10 days late is no longer accepted. If you submit too close to the end of the semester for a particular assignment, you will also receive no points or credit. If you are late due to extenuating circumstances, we will consider exceptions on a case-by-case basis if appropriate documentation is provided.

Lecture Schedule (Tentative)

Wk	Date	Instructor	Lecture Topic	Deliverables
1	Jan 13th	Jae	● Course Logistics ● Introduction: “What is Software Engineering?”	None
2	Jan 20th	(No lecture) MLK Day Holiday		Startup (Team Charter) formation and project proposal
3	Jan 27th	Mahdi	● What causes software to fail? ○ History of software failures ○ Enterprise vs Startup	Project plan and backlog creation
4	Feb 3rd	Jae	● Software engineering philosophy ● Incremental commitment spiral model ● Agile	Sprint 1: Deliver prototype requirements and backlog
5	Feb 10th	Mahdi	● Agile: State-of-Industry	Sprint 2: Risk analysis and feasibility report
6	Feb 17th	(No lecture) President’s Day Holiday		Sprint 3: Review and Git repository setup
7	Feb 24th	Jae	● Analysis and planning ● Software architecture design ● Team Presentation #1	Sprint 4: Architecture review and mockups
8	Mar 3rd	Mahdi	● Evaluation Metrics & Management	Sprint 5: Review and Git repository setup
9	Mar 10th	Jae	● Prototyping UI/UX ● Reviews and testing ● DevOps ● Team Presentation #2	Sprint 6: First feature prototype delivery
10	Mar 17th	(No lecture) Spring Recess		
11	Mar 24th	Jae	● Software maintenance ● Software engineering best practices ● Exam #1 Preparation ● Team Presentation #3	None

12	Mar 31st	Exam #1 (Location and time TBD)		
13	Apr 7th	Mahdi	● Cloud Nativeness	Sprint 7: Microservices
14	Apr 14th	Jae	● Software ethics ● Project management ● Team Presentation #4	None
15	Apr 21th	Mahdi	● AI is a commodity now	MVP Delivery - D-Day
16	Apr 28th	Jae	● Future of software engineering ● Exam #2 Preparation ● Team Presentation #5	None
17	May 5th	(No lecture) Study week		
18	May 12th	Exam #2 (Location and time TBD)		

Academic Integrity

Unless otherwise noted, this course will follow the expectations for academic integrity stated in the USC Student Handbook. The general USC guidelines on Academic Integrity and Course Content Distribution are provided in the subsequent “Statement on Academic Conduct and Support Systems” section.

Students must work independently on all individual assignments; collaborating on individual assignments is considered cheating and will be penalized accordingly. All USC students are responsible for reading and following the USC Student Handbook, which prohibits plagiarism. Some examples of behavior that is not allowed are copying all or part of someone else’s work (by hand or by looking at others’ files, either secretly or if shown) and submitting it as your own; giving another student in the class a copy of your assignment solution; consulting with another student during an exam; and copying text from published literature without proper attribution. If you have questions about what is allowed, please discuss it with the instructor.

Students who violate University standards of academic integrity are subject to disciplinary sanctions, including failure in the course and suspension from the university. Since dishonesty in any form harms the individual, other students, and the university, policies on academic integrity have been and will be strictly enforced.

Please ask the instructor and/or TA(s) if you are unsure about what constitutes unauthorized assistance on an exam or assignment or what information requires citation and/or attribution.

You may not record this class without the express permission of the instructor and all other students in the class. Distribution of any notes, recordings, exams, or other materials from a university class or lectures — other than for individual or class group study — is prohibited without the express permission of the instructor.

Use of Generative AI in this Course

You are expected to use AI (*e.g.*, ChatGPT and image generation tools) in this class. Learning to use AI is an emerging skill; this is an opportunity for you to discuss with the instructor the appropriate use of these tools.

Remember the following:

- AI tools are permitted to help you brainstorm topics or revise work you have already written.
- If you provide prompts that require minimum effort, you will get low-quality results. You will need to refine your prompts to get good outcomes, and this will take work.
- Proceed with caution when using AI tools, and do not assume the information provided is accurate or trustworthy. If it gives you a number or fact, assume it is incorrect unless you either know the correct answer or can verify its accuracy with another source. You will be responsible for any errors or omissions provided by the tool. It works best for topics you understand.
- AI is a tool, but one that you need to acknowledge using. Please include a paragraph at the end of any assignment explaining if, how, and why you used AI and indicate/specify the prompts you used to obtain the results. Failure to do so is a violation of academic integrity policies.