

Introduction to Software Engineering

COURSE SYLLABUS

(2025 Fall)

PROFESSOR: Eunseok Lee (이은석)

OFFICE:	Building of Corporate Collaboration Center, Room 85457(TA's office)/85451(Dean's office)
OFFICE HOURS:	4:30-5:30 MON & WED; 1:00-3:00 THR; 4:30-5:30 FRI
OFFICE PHONE:	031-290-5511 (mobile, 010-4751-8833)
E-MAIL ADDRESS:	leees@skku.edu
WEB PAGE:	http://selab.skku.ac.kr
COURSE PAGE:	http://canvas.skku.edu (과목/강의콘텐츠)
CLASS HOURS:	

A. DESCRIPTION

This course involves an overview of modern software engineering techniques. The techniques correspond to the software's life cycle and range from requirement engineering to testing, covering imperative knowledge for software engineers. The comprehensive knowledge is classified into three groups of issues. Each group covers the following:

- Basic concepts and notions on SE
- Fundamental knowledge on main activities and work products during the process of SE
- Global standards and trends on SE like Agile method, UML, CMMI and SPICE

This course originally aims to comprehensively cover the entire process from requirements gathering to design, implementation, and testing. Especially this semester, it is highly recommended to actively utilize LLMs like ChatGPT to carry out these processes. However, it is necessary to explicitly explain how the LLM was utilized.

B. ORGANIZATION

This course will be conducted in the form of a flipped class, utilizing the lecturer's online lectures (referred to as **pre-class**) and offline classes (referred to as **in-class**). During the offline classes, attendees will answer Q&A sheets provided for each session, and there will be a couple of presentations for team projects. For the lectures, one textbook will be used along with several related books and technical papers. Additionally, students will be required to write Q&A sheets for each chapter and take comprehensive exams, such as midterm and final exams.

***Flipped Class:** It is a switched version of lecture and application.

A flipped class can be divided into pre-class and in-class. The pre-class is an online class, and students should study the lecture video provided by the lecturer in advance before the offline class. During the in-class session, all students should individually fill out the Q&A sheet or discuss it with their team members, utilizing the knowledge learned from the pre-class. Besides the questions that the lecturer provides, each student must create one or more questions for each chapter. Consequently, students must write an answer sheet for the questions provided by the lecturer and the questions they create. In special circumstances where social distancing is required, such as during the COVID-19 pandemic, the offline class can be replaced with a real-time streaming class. Failing to study the pre-class material in advance may prevent you from effectively participating in the in-class activities and may also disrupt others' learning experiences.

"The best way to communicate with me is through email at leees@skku.edu."

C. COURSE OBJECTIVES

1. To provide students with limited knowledge and experience an introduction to essential basic concepts in software engineering. (Engineers with Basic SE Techniques)
2. To instill in students a strong sense of ethical and professional responsibility as software engineers. (Ethical Engineers)
3. To familiarize students with current emerging issues in software engineering techniques, such as UML, agile methods, TDD (test-driven development), SPICE, and CMMI. (Updated Engineers)
4. To enable students to gain practical experience by working on small-scale projects in teams and applying the SE techniques they have learned. (Interactive & Collaborative Engineers)
5. To equip students with the ability to be competitive engineers, capable of

selecting the best practices and making informed decisions in engineering processes. (Smart Engineers)

D. COURSE TOPICS

The course will cover the following topics:

1. Introduction to Software Engineering
2. System Engineering
3. Software Processes
4. Agile Software Development
5. Requirement Engineering
6. UML(Unified Modeling Language)
7. System Modeling
8. Architectural Design
9. OMT(Object Modeling Technique)
10. Project Management
11. Project Estimation
12. Design and Implementation
13. Software Testing
14. Software Process Assessment and Improvement(optional)

E. TEXT AND Related Papers

1. Software Engineering (10th), Ian Sommerville 2021.
2. Software Engineering - a practitioner's approach (9th). R. Pressman. McGrawHill 2018.
3. Other SE related papers and reviews

F. GRADING PLAN

Coursework will be weighted as follows:

1. Team project	25%
2. Q/A sheet	20%
3. Midterm exam	25%
4. Final exam	25%
5. Attendance	<hr/> <hr/> <u>5%</u>
	25% + 70% + 5% = 100%

G. TEAM PROJECT

The main purpose of the team project is to provide students with hands-on

experience in the software development process while fostering problem-solving skills through effective communication and collaboration among team members. The team project will span the entire semester, and the themes will vary each semester to align with the current needs and trends of the era.

Participants will be expected to utilize trending technologies, including AI, LLM, machine learning techniques, to carry out their projects.

To facilitate the process, it is recommended that students first find a reference site and then modify and enhance it with their unique ideas. Existing techniques can be borrowed and incorporated into the projects. All project activities and outcomes should be transparent and managed through Github. Each student must create a Github ID to commence their project work. Utilizing at least three open-source software packages for the project is strongly encouraged.

Each team will be required to produce various documents, such as proposals, requirements specifications, design specifications, code and test case, in accordance with provided guidelines. Detailed evaluation methods and criteria will be introduced later. Under ordinary circumstances, each team will consist of four or five members.

H. MIDTERM and FINAL EXAM

The midterm and final exam will be conducted to confirm your comprehensive knowledge on the course content.

I. ATTENDANCE

Attendance will be graded as follows:

No absences:	5
One absence:	4
Two absences:	3
Three absences:	2
Four or more:	F

Absences for which a medical or court excuse is provided (on a professional letterhead) will be recorded but will not affect the attendance grade. Similarly, one absence for which advance notice is given via email or in person will not be counted towards the attendance grade. However, any significant tardiness or early departure from class will be considered an absence.

It is important to note that students with more than four unexcused, full-class absences will automatically receive an "F" grade for the COURSE, as per the College's rules.

J. TENTATIVE SCHEDULE

Weeks	TOPIC/ACTIVITY
Week 1	Orientation
Week 2	Introduction to software engineering
Week 3	System engineering
Week 4	Software processes
Week 5	Proposal presentation of team project
Week 6	Agile software development
Week 7	Requirements engineering
Week 8	Midterm exams
Week 9	System modeling / OMT(Object Modeling Technique)
Week 10	Architecture design
Week 11	Design and implementation
Week 12	Project management / Estimation
Week 13	Software testing
Week 14	Software process assessment and improvement
Week 15	Final presentation of team project
Week 16	Final exams