



King Mongkut's University of Technology Thonburi
Department of Computer Engineering
Faculty of Engineering

Course Syllabus

Last update: 2 August 2022

Course Title	CPE 327 Software Engineering
Course Credit	3 Credits
Semester/Year	1/2022 (1/2565)
Prerequisite	CPE100 Introduction to Computer Programming
Schedule	Lectures Thai sections - Monday 10:30-12:30 International sections – Monday 13:30-15:30 Labs Section D (Inter) – Tuesday 08:30-10:30 Section C (Inter) – Tuesday 10:30-12:30 Section A (Thai) – Tuesday 13:30-15:30 Section B (Thai) – Tuesday 15:30-17:30 Residential College – Tuesday 15:30-17:30
Instructors	Dr. Phond Phunchongharn Email: phond.p@mail.kmutt.ac.th Dr. Khajonpong Akkarajitsakul Email: khajonpong.akk@mail.kmutt.ac.th
Teaching Assistants	Thai Sections Suthawee Weraphong (P'Mek) suthawee.wera@kmutt.ac.th Pataraphorn Tanutsiriteeradet (P'Mint) pataraphorn.mintpat@mail.kmutt.ac.th RC Section Niti Buesamae (P'Fluke) niti.bue@mail.kmutt.ac.th International Sections Nutaya Pravalphruekul (P'Pear) nutaya.pr63@mail.kmutt.ac.th Arnatchai Techaviseschai (P'Art) arnatchai.tec63@mail.kmutt.ac.th

Course Description

This course introduces students to principles and techniques used to create functionally correct, easy to use, robust, reliable, and maintainable software systems. The course covers all phases of the software development lifecycle, focusing on practical approaches that can be applied in each phase. Lectures are supplemented by homework assignments to encourage problem-based learning. The course also includes a team-based collaborative term project that requires students to analyze a proposed software system and produce a set of development artifacts typical of a real-world software development project.

Learning Outcomes

By the time students have concluded this course, they should be able to:

1. List and explain the objectives of using a software engineering approach to software development.
2. List and explain the key activities involved in a software engineering approach, including requirements analysis, design and modeling, validation and verification, configuration management, project management and monitoring, etc., and discuss how these different activities interact and influence each other.
3. Explain the concept of a software engineering process framework. Compare and contrast well-known process frameworks including waterfall, Rational Unified Process, eXtreme Programming and Scrum.
4. Apply a team-defined set of software engineering processes in developing a team software project of moderate complexity.
5. Analyze the use of software engineering processes in a team software project, suggesting areas or activities where the processes could have been improved.

Teaching methods: Lectures and problem-based learning

Course evaluation:

• <i>Labs and Homework</i>	<i>20%</i>
• <i>Term Project</i>	<i>30%</i>
• <i>Term Project Progress and Discussion</i>	<i>10%</i>
• <i>Quiz (4 quizzes)</i>	<i>10%</i>
• <i>Midterm</i>	<i>15%</i>
• <i>Final</i>	<i>15%</i>

References

1. Software Engineering: A Practitioner's Approach. 8th Edition. Roger S. Pressman. McGraw-Hill. 2015.
2. Software Engineering, 9/E. Ian Sommerville, University of St Andrews, Scotland, Addison-Wesley.

Policy regarding cheating

The honesty policy will be strictly enforced. Any assigned work, if copied with permission, all persons involved will receive **a negative score** equivalent to the full score of the assigned work for first violation; a second violation will result in **F** for the course.

Class Schedule

Subject to change

	Topics	Labs/Exercises	Lecturer	Date
1	Introduction to software engineering: objectives and activities	• Critique a software project you created in a previous course	P	Aug. 8
2	Software process frameworks overview	• RUP to Scrum conversion exercise Students choose a project topic	P	Aug. 15
3	Requirements analysis/problem definition – Use Case analysis & Use Case diagram	• Use case modeling exercise	M	Aug. 22
4	Functional design and usability –	• UI design exercise with Figma • Sequence diagram exercise	M	Aug. 29
5	Architecture and detailed design – UML component, class, state diagrams	• Class diagram exercise • State diagram exercise	M	Sep. 5
6	Project Progress and Consult 1 (10 mins each)			Sep. 12
7	Software configuration management	• Git/Github exercise Students submit first project document	M	Sep. 19
8	Software implementation: strategies for quality	• Refactoring exercise	M	Sep. 26
9	Software validation and verification	• Test plan development exercise • Junit exercise	M	Oct. 3
10	Agile in Depth (invited speaker)	• Agile workshop with Jira	P	Oct. 10
11	Quality Assurance (invited speaker)	• QA Lab	P	Oct. 17
12	Midterm Exam (9.00-11.00)			Oct. 25
13	Project Progress and Consult 2 (15 mins each)			Oct. 31
14	Managing software teams (invited speaker)	• MS Project workshop (using the students' own project) Students submit second project document	P	Nov. 7
15	Dev Ops (invited speaker)	• Continuous Integration (CI) Lab	P	Nov. 14
16	Process maturity and continuous improvement (CMM/CMMI)	<i>Live project demos or video demos</i>	M	Nov. 21
17	Platforms, architectures and software process	<i>Students submit final project document and code.</i>	P	Nov. 28
18	Final Exam (9.00-11.00)			Dec. 6
19	Project Final Presentation (15 mins each)			Dec. 15