

Software Engineering

Course Name	Course type (credit/hours)	Elective course(3/3)		Course code	F001
	Target students Division/major/grade	Software and Computer Engineering/Junior		Opening semester	2019 2ND SEMESTER
	Class time and classroom	Mon B(Pal407)Thu B(Pal407)		English Grade	A(100%English)
Reference to this course	Prerequisite courses				
	Related basic courses				
	Recommended concurrent courses				
	Related advanced courses	Advanced Software Engineering (Masters Course)			
Instructor	Name (title/division)		Seok-Won Lee(Professor, Software and Computer Engineering)		
	Office Room Number	팔달관 603호	Office phone Number	3548	e-mail
	Office hours	Thursdays 3:00-4:30PM (appointments)		Homepage address	http://www.machinediscovery.com
Teaching Assistant	Name (title/division)				
	Office Room Number	팔달관 913-1 (NiSE 연구실)	Office phone Number	2442	e-mail

1. Introduction

This course introduces to the student the basic concepts, principles, and dynamics of software engineering. It involves the study of methodologies and technologies, and the construction of models at each major software development phase, namely, Requirements and Analysis, Design, Implementation, and Testing. It also discusses the methodologies and technologies of software quality assurance and change management. There is a project assignment in this course to enhance the practical skill of software engineering.

2. Course Objectives

* Objectives:

- Students will learn about many aspects of working with a team on the projects to produce quality software products on time and within budget.
- The student will gain an appreciation of the tools and techniques used to develop software systems within a group context.

* Topics to be studied include:

- life cycle models,
- functional specification and design of real-time systems,
- testing strategies and quality management

* Accomplishments:

By completing this course, the students will be able to:

- 1) apply techniques for control of software project
- 2) construct and validate a software specifications and products
- 3) functional design of software systems
- 4) describe software systems using appropriate language and technical specification techniques to suit the intended audience

3. Class types and activities

- "1) Lecture
- 2) Team Project
- 3) Project Reports and Oral Presentations"

4. Teaching Method

- | | |
|---|---|
| <input checked="" type="checkbox"/> lecture | <input checked="" type="checkbox"/> discussion and debate |
| <input checked="" type="checkbox"/> team project(presentation and case studies) | <input type="checkbox"/> experiments(role-playing,etc) |
| <input checked="" type="checkbox"/> designing and production | <input type="checkbox"/> on-site learning(on-site training) |
| <input type="checkbox"/> others | |

5. Support Systems in Use

- | | | |
|--|---|--|
| <input checked="" type="checkbox"/> AjouBb | <input type="checkbox"/> automatic recording system | <input checked="" type="checkbox"/> web-based assignment |
| <input type="checkbox"/> cyber lecture | <input type="checkbox"/> online content | |
| <input type="checkbox"/> class behavior analyzing system | <input type="checkbox"/> others | |

6. Teaching Tools

- | | | |
|--|---|--|
| <input type="checkbox"/> PBL(Problem Based Learning) | <input type="checkbox"/> CBL(Case Based Learning) | <input checked="" type="checkbox"/> TBL(Team Based Learning) |
| <input type="checkbox"/> UR(Undergraduate Research) | <input type="checkbox"/> FL(Flipped Learning) | <input type="checkbox"/> DSAL(Data Science Active Learning) |
| <input type="checkbox"/> others | | |

7. Knowledge and ability required for taking this course

"Students should have:
-familiarity with the commonly used software analysis and design methodologies, including structured analysis and design and the object oriented techniques;
- basic programming skills and general familiarity with the processes involved in the development of software using languages like Java and C/C++."

8. Method of Evaluation

Evaluation Item	The Number of Times	Evaluation Proportion	Remarks
Attendance		10%	
midterm exam		20%	
final exam		30%	
quiz		10%	
presentation		10%	project proposal, mid, final term project presentation
discussion			
homework		20%	team project
etc			
study hours			

9. Textbook and supplementary material

Main/Sub	Title (Web-site)	Writer	Publisher	Publication year
Main	Software Engineering 9th Ed.	Ian Sommerville	Pearson	2011
Sub	Software Engineering A Practitioner's Approach 7th Ed.	Roger Pressman	McGraw-Hill	2010

10. Class system and Class shedule

"Course Contents are:

1. Introduction to Software Engineering
2. Software Process – Generic view of process, Process models, Agile process
3. Software Engineering Practices Requirement Engineering, Design Engineering, Testing Strategies
4. Managing Software Projects Quality management, Configuration Management"

*** Team project is mandatory (usually it consists of 3 –4 team members).

It is strongly encouraged to hold at least 1–2 outside classroom project team meetings to promote the team communication skills.

Also, there will be 1 –2 project presentations based on the team project work progress.

< Class Schedule >

* language : K-korean, E-English

Weeks	Topics	language	Instructor	Teaching Method	Evaluation Method	Matter to be prepared
1	Software and Software Engineering	E	Seok-Won Lee	lecture		
2	software process	E	Seok-Won Lee	lecture		
3	agile process	E	Seok-Won Lee	lecture		
4	requirements engineering	E	Seok-Won Lee	lecture		
5	system modeling	E	Seok-Won Lee	lecture		
6	architectural design	E	Seok-Won Lee	lecture		
7	design process	E	Seok-Won Lee	lecture		
8	mid term exam	E	Seok-Won Lee	test		
9	impementation process	E	Seok-Won Lee	lecture		
10	demo 1	E	Seok-Won Lee	lecture		
11	software quality management	E	Seok-Won Lee	lecture		
12	software testing	E	Seok-Won Lee	lecture		
13	code inspection	E	Seok-Won Lee	lecture		
14	demo 2	E	Seok-Won Lee	lecture		
15	code inspection	E	Seok-Won Lee	lecture		
16	final exam	E	Seok-Won Lee	test		

11. Other items of notification

This is the prerequisite class for CSE645, Advanced Software Engineering (Masters program)