1. Course Title (Course Code)

Software Engineering (2236)

2. Instructor

Mamoru ITO

3. Term

Spring 1

4. Outline and Objectives

Software plays an increasingly important role in the evolution of ICT systems. However, it is actually hard to develop software on time, on budget, and on target. It is widely believed that many software development projects fail or are challenged. This course will provide comprehensive and interdisciplinary learning opportunities for those who wish to tackle with the challenges of software projects. This course will cover both software engineering and project management issues.

5. Goals (Attainment Targets)

(1) To have the basic understanding of software development life cycle and process models
(2) To be able to utilize basic techniques in software analysis and design
(3) To acquire the practical thinking and decision-making skills required for software project management
(4) To deepen an understanding of social environments surrounding software development

6. Correspondence relationship between Educational goals and Course goals

<table>
<tr><th colspan="3">Educational goals</th><th>Course goals</th></tr>
<tr><td rowspan="2">High level ICT skill</td><td colspan="2">Basic academic skills</td><td>(1), (2)</td></tr>
<tr><td colspan="2">Specialized knowledge and literacy</td><td>(1), (2)</td></tr>
<tr><td rowspan="8">Human skill
(Tankyu skill)</td><td colspan="2">Ability to continually improve own strengths</td><td>(2)</td></tr>
<tr><td rowspan="4">Ability to discover and resolve the problem in society</td><td>Problem setting</td><td>(3), (4)</td></tr>
<tr><td>Hypothesis planning</td><td>(3), (4)</td></tr>
<tr><td>Hypothesis testing</td><td></td></tr>
<tr><td>Practice</td><td></td></tr>
<tr><td rowspan="3">Fundamental Competencies for Working Persons</td><td>Ability to step forward</td><td>(3)</td></tr>
<tr><td>Ability to think through</td><td>(3), (4)</td></tr>
<tr><td>Ability to work in a team</td><td>(1), (2)</td></tr>
<tr><td colspan="3">Professional ethics</td><td>(3), (4)</td></tr>
</table>

7. Course Requirements

None

8. Textbooks

None

9. Reference Books

Pressman, Roger S. *Software Engineering: A Practitioner's Approach*. New York: McGraw-Hill Higher Education, 2010.

10. Evaluation

Goals	Evaluation method & point					
	term-end exam	quiz	report	presentation	deliverable	other
(1)	○	○		○	○	
(2)	○	○		○	○	
(3)	○		○	○	○	
(4)	○		○			
Allocation	30	25	25	10	10	

11. Notes

This course does not use a textbook, but course materials will be provided on Moodle.

Course Schedule
(Notice) This schedule is a tentative plan, there might be changes, additions, revisions, etc. at the time of delivering the course.

Lesson 1 Basic Principles (Lecture: 90 min)

The Software runs on a computer. The basic knowledge on computers expands an understanding of software development. We will learn how a computer works in this lesson.

1. Introduction
2. Computer organization
3. Machine language and assembly language
4. Flow of a program
5. Interrupt handling

Lesson 2 Data Structures and Algorithms (Lecture: 90 min)

The data structures and algorithms should be considered for us to design a computer program. This lesson will provide students with the introduction of data structures and algorithms.

1. Major data structures – array, list, stack, queue, and tree
2. Major algorithms - sorting algorithms and search algorithms
3. Computational complexity

Lesson 3 Software and Software Engineering (Lecture: 90 min)

The software grows increasingly important along with the popularization of computers. We will discuss the reality surrounding software development after understanding of the features of the software and learn the necessity of software engineering.

1. Features of software
2. Importance of software
3. Environment surrounding software development
4. Role of software engineering

Lesson 4 Software Development Processes (Lecture: 90 min)

A “Process” can be defined as a “set of interrelated or interacting activities, which transforms inputs into outputs”. Good process is required to produce good outputs. We will learn the overview of software life cycle process models and the meaning of process improvement.

1. Definition of software process
2. Life cycle models
3. Present situation and issues on software process
4. Meaning of software improvement

Lesson 5 Project Management Processes (Lecture: 90 min)

Generally, software is developed by a project team. The project team should be managed adequately. This lesson will clarify a project, project management, project lifecycle, and project organizations.

1. Definition of project and project management
2. Project life cycle
3. Relationship with organizations and stakeholders
4. Trend in project management standards

Lesson 6 Requirements Analysis (Lecture: 90 min)

The role of software engineer is to realize the requirements of customers and users by use of software. But their requirements are sometimes ambiguous and lack consistency. We should acquire their requirements exhaustively and analyze them systematically. We will marshal the concepts of requirements and flow of requirement analysis.

1. Difference between need want and demand
2. Functional requirements and non-functional requirements
3. Flow of requirements analysis
4. Requirements modeling

Lesson 7 Software Design (Lecture: 90 min)

The optimum design technique should be selected based on the target and objectives of software development. This lesson introduces major software design techniques such as structured design and object-oriented design.

1. Introduction of major software design techniques
2. Structured design
3. Object-oriented design

Lesson 8 Software Testing (Lecture: 90 min)

Software testing is becoming important because defects in software have the significant impact on the society. We will learn the positioning of software testing, kinds of software testing, and testing techniques in this lesson.

1. Necessity and limitation of software testing
2. Flow of software development and testing phases
3. White box test and black box test
4. The major testing techniques

Lesson 9 Software Quality (Lecture: 90 min)

One of the objectives of software engineering is to develop high-quality software. Management on software quality is more important than that of hardware quality because software is invisible. We will understand the whole picture of software quality and necessary activities to achieve the required quality in this lesson.

1. Difference between quality and grade
2. Software quality model
3. Quantitative quality management
4. Software design review

Lesson 10 Object-Oriented Methodology (Lecture: 90 min)

Object-oriented methodology is becoming popular in association with increasing in size and complication of software. This methodology is used not only for programming but also for requirements analysis and software design. This lesson will focus on object-oriented analysis and design by the use of UML modeling.

1. A brief history of object-oriented methodology
2. Object-oriented analysis
3. Modeling and UML diagrams
4. Object-oriented design

Lesson 11 Unified Modeling Language (UML) Part 1 (Lecture: 90 min)

UML is becoming commonly-used with object-oriented technology. UML stands for Unified Modeling Language, which is a useful tool for analysis and design of complex software systems. We will learn how to describe major diagrams. This lesson focuses on the structural diagrams: Class Diagram and Object Diagram.

1. Overview
2. Modeling types and standard diagrams
3. Class Diagram
4. Object Diagram

Lesson 12 Unified Modeling Language (UML) Part 2 (Lecture: 90 min)

UML is becoming commonly-used with object-oriented technology. UML stands for Unified Modeling Language, which is a useful tool for analysis and design of complex software systems. We will learn how to describe major diagrams. This lesson focuses on the behavioral diagrams: Use Case Diagram, Sequence Diagram, State Machine Diagram, and Activity Diagram.

1. Use Case Diagram
2. Sequence Diagram
3. State Machine Diagram
4. Activity Diagram

Lesson 13-14 Exercises in Analysis and Design (Exercises: 180 min)

Software analysis and design includes various activities from requirements analysis to implementation, which holds extremely important position in software developments. In the following three lessons, exercises in the analysis and design of software systems are conducted through group work. After the exercises, each group of the students makes a presentation on the results of group work.

1. Exercises in structural and behavioral modeling
2. Exercises in analysis and design of software
3. Exercises in drawing UML diagrams
4. Presentations

Lesson 15 Summary and the Latest Topics (Lecture: 90 min)

Software engineering is advancing as well as information and communication technologies. We will learn the latest trend in software engineering. Issues in engineering ethics also will be covered in this lesson.

1. The latest topics of software development
2. The latest topics of project management
3. Engineering ethics in software

Lesson 16 Term-end Examination (Exam: 90 min)

The term-end examination will be conducted to evaluate an achievement degree of each student.