



UNIVERSITAS
INDONESIA

Veritas, Probitas, Iustitia

FACULTY OF
**COMPUTER
SCIENCE**

COURSE PROFILE

CSC3602205
Software Project
(Proyek Perangkat Lunak)

2017

ACKNOWLEDGMENT AND CHANGE NOTES

Version 1.3 – December 2017 – Fasilkom UI

This version is updated by Ade Azurat and Maya Retno Ayu S from the previous version. There are several minor changes with respect to more detailed or revised issues. Changes highlights are: Weekly Monitoring, Weekly Evaluation Sheet, Assistant role, Industry engagement and evaluation mechanism.

Version 1.2 – January 2016 – Fasilkom UI

This version is updated by Ade Azurat, Maya Retno Ayu S and Niken Fitria Apriani from the previous version. There are several minor changes with respect to more detailed or revised issues. Changes highlights are: Assistant's role and evaluation mechanism. Typesetting and cover page are modified and designed by Dalilah Ghaisani

Version 1.1 – December 2015 – Fasilkom UI

This syllabus is updated by Ade Azurat and Niken Fitria Apriani from the previous version composed by Eko K. Budiardjo, Hadaiq Rolis S., Nurul Qomariyah, and Satrio B. Yudhoatmojo. There are huge changes to accomplish industry-university partnership. Lay out is designed by Anggiearanidipta Suma M. and Team 3200 Fasilkom UI. Cover page illustration is designed by Ade Azurat.

SECTION 1 : GENERAL INFORMATION

Course Name	:	Software Project (Proyek Perangkat Lunak - PPL)
Course Code	:	CSC3602205
Intended for	:	Third year (Semester 6) students
Semester Credit Hours	:	6 SKS (equivalent to 17 hours per week work load)
Weekly Credit	:	3 hours
Implementation	:	Reporting <ul style="list-style-type: none">• Individual weekly report preparation• Class Room 11 hours Project works <ul style="list-style-type: none">• 10 hours - Project works (based on backlogs, including: code, design, writing, etc)• 1 hour - Team meeting (supervised by assistants) 3 hours Self-study <ul style="list-style-type: none">• tools adaptation• new technology exploration
Prerequisites and Related Courses	:	Software Engineering Basic Programming Skill and Data Structures Web Programming and Design Database Computer Network
Lecturer	:	Dr. Eko K Budiardjo < eko@cs.ui.ac.id > Dr. Ade Azurat < ade@cs.ui.ac.id > Bayu Anggorojati < bayuanggorojati@cs.ui.ac.id > Gladhi Guarddin < adin@ui.ac.id > Maya Retno Ayu S < mayaretno@cs.ui.ac.id >
Teaching Assistant	:	To be announced
Course Description	:	This course presents a complete software development life cycle, from ideation until deployment. Issues related to development methodology and technology used will be discussed.

SECTION 2: LEARNING OBJECTIVES

MAIN GOALS

1. Students are able to plan, manage, implement and evaluate an IT Project.
2. Students are able to apply standard techniques of software engineering and IT Project Management.
3. Students are able to practice good soft skill of software engineer.

INTERMEDIATE GOALS

1. Students are able to choose and perform steps of IT Project Management correctly.
2. Students are able to implement IT Project using a certain framework and tools.
3. Students are able to practice teamwork effectively.
4. Students are able to perform a complete software development life cycle.
5. Students are able to practice communication skills either orally or written as IT Professionals.

COMPETENCE TOPICS

1. Working in team, communication skill
2. Product visibility, ideation
3. Development and deployment, continuous integration
4. Software testing, test coverage
5. Maintainability, refactoring, changes management
6. Security, privacy
7. Documentation
8. Scalability, profiling

The competences as software engineering are developed based on:

- *Software Engineering Body of Knowledge (SWEBOK), IEEE Computer Society, 2014*
- *Software Engineering Competency Model (SWECOM), IEEE Computer Society, 2014*

COURSE REFERENCE

- Text book : [1] Pressman, Roger S. Software Engineering: A Practitioner's Approach 6th ed. McGraw Hill, Singapore: 2005.
- [2] Dennis, Allan. System Analysis and Design with UML: An Object-Oriented Approach, 3rd ed. John Wiley & Son, Asia: 2010.

SECTION 3: PROGRAM MAPPING

Week	Weekly Activities	Milestone/ Deliverables	Class Activities		Submission by Student
			Session 1	Session 2	
1	Introduction, Brainstorming, review idea	Topics	Lecture by Dr. Eko K Budiardjo: Brainstorming	Workshop by Partners: Ideation	
2	Write proposal	Proposal	Team Presentation: Product Proposal (incl. mock-up)		Proposal: Mon: feedback; Wed: graded by TA
3	Design wireframe	Wireframe + flow	Lecture by Dr. Eko K Budiardjo: scrum, git	By Partner+ TA, Lecturer: Simulasi Scrum + git	Wireframe: Mon: graded by TA
4	Sprint plan 1	Product backlog	by Lecturer+ TA: Team Review: Product backlog, Sprint Planning		backlog on Trello; Mon: Feedback; Wed: graded by TA
5	Development sprint 1	Standard Working environment	Individual Weekly Progress		
6	Development sprint 1	Git progress, code review	Lecture by Partner: Clean Code, Microservices	Lecture by Partner: Unit Testing, Refactoring, CI	
7	Development sprint 1	Git progress, code review	Individual Weekly Progress		
8	Sprint review 1, sprint plan 2	Executable prototype & Document	Team Review and Presentation		Mid-Term replacement, Mon: Progress Presentation slide
9	Development sprint 2	Git progress, code review	Individual Weekly Progress		
10	Development sprint 2	Git progress, code review	Lecture by Partner: Security, Deployment	Lecture by Partner: Scalability, Profiling	
11	Development sprint 2	Git progress, code review	Individual Weekly Progress		
12	Sprint review 2, sprint plan 3	Publishable product & Document	Team Review and Presentation		Mon: Presentation + deployed on server/cloud/store
13	Development sprint 3	Git progress, code review	Individual Weekly Progress		
14	Development sprint 3	Git progress, code review	by selected teams: Technology Sharing	Lecture by Dr. Eko K Budiardjo: Monetizing, Pitching	
15	Development sprint 3	Git progress, code review	Individual Weekly Progress		Mon: individual blog, and activity in git or trello
16	Sprint review 3, deployment	Publishable Product & Document	Team Review and Presentation		Final Term replacement, Mon: Pitch Deck, Video Demo
Partner Evaluation and Feedback during Product exhibition on Summer camp (http://summercamp.cs.ui.ac.id/) on June 2017					

Lecturers and the students can choose on methodology to follow to simplify class arrangement. The minimum constraints of the program mapping are:

- (a) There should be a minimum of 3 weeks of ideation. Most likely in the beginning.
- (b) There should be at least one product/project review or presentation. Most likely in the ending.
- (c) There should be at least one milestone or deliverable for each sub-tasks:
 - a. Evidence of source code repository (e.g. gitlab, github, mercurial)
 - b. Evidence of project management and communication (e.g. trello, jira)
 - c. Evidence of design (e.g. API, wireframe, data model)
 - d. Evidence of requirement (e.g. product backlog, use case)
 - e. Evidence of testing or evaluation (unit test, profile)
 - f. Evidence of deployment (e.g. play store, CI server/configuration)

INDUSTRY ENGAGEMENT

- Partners who would like to suggest topics should submit topics descriptions by 10 January 2017
- Partners are expected to attend ideation Workshop as mentor to help the students understand and build the idea. Ideation will be held on second week of February 2017 (exact date will be given by 16 January 2017)
- Partners are expected to attend product exhibition as reviewers who evaluate all students' products. Product exhibition will be part of Summer Camp (<http://summercamp.cs.ui.ac.id/>) event which usually be held on June every year.
- Partners are invited to give presentation on selected course subjects. List of subject and presentation date, will be available by 16 January 2017.

WEEKLY ACTIVITY

Each week, starting from week 2nd, students should:

- (a) Work on their own tasks as part of the team
- (b) Self study, learn and explore the latest technology
- (c) Create a short weekly progress notes.
 - a. The notes should be made accessible by the lecturers, partners, and assistant.
 - b. The students are expected to spend at least 30 minutes to write the notes before the weekly meeting with lecturer (point (c))
 - c. The expected length of the notes is 200 – 500 words.
- (d) Attend min the 1 hour weekly team meeting to do:
 - a. Team work and discussion, supervised by Assistant.
 - b. Daily Scrum Meeting
 - c. Preparation for Sprint planning or Sprint Review (if applicable)

SECTION 4: EVALUATION

GRADING SCHEME

20 %	Progress by teaching assistants
40 %	Bi-Weekly Skill/Knowledge achievement and report by Lecturer
20 %	Project Contribution (peer, TA, lecturer review)
20%	Final Product Quality by Partners (during summer camp exhibition)

EVALUATION CRITERIA

1. Weekly individual progress notes and knowledge sharing.
2. Contribution in team work
3. Quality of work and team work
4. Tool proficiency
5. Time management, punctuality of report/deliverables
6. Knowledge, skill, understanding and application in project.
7. Team and individual improvement
8. Satisfaction or Feedback from client or partner

INDIVIDUAL BI-WEEKLY PROGRESS EVALUATION BY LECTURER:

Students are able to monitor the score of their bi-weekly progress evaluation by lecturer. It is consisted of two parts and three sub-evaluation.

Part A is related to the knowledge and skills. Student should prove their understanding or skill to the mentor and lecturer by showing concrete evidence and academic justification. It can be communicated textually or orally. Once it has been evaluated and has been graded, it will stay the same until the student improve it. There are **mandatory** and **elective topics**. Those topics are grouped based on generic software engineering phases. The **mandatory topics** are prescribed. All students should learn and master all of mandatory topics. The number of **elective topics** is defined but the exact topics are left to the students to choose. The final score of part A is the average of all topics. The score of each topic is the score of the last bi-weeks or the highest one.

Part B is related to behavior attributes or soft skill as indicated by the references (SWECOM 2014). All students are monitored and graded every week. The final score for part B is the average score of all weeks.

Each Bi-Weeks Score is the score taken from the average of part B that bi-weeks, and the highest 3 topics of Part A that are evaluated on that bi-weeks.

The score convention of the student understanding, skill or attitude is as follows:

- 0 (E) : No proof or no show. The student does not do, nor know it.
- 1 (D) : unacceptable, repeating or memorizing without understanding
- 2 (C) : acceptable, but below expectation, or below average
- 3 (B) : as expected or average
- 4 (A) : above average, exceed expectation

The final score for progress component by lecturer is the average of Part A, Part B, average of each week score. In the next page is an example of individual progress evaluation form based on the mentioned references. Students are expected to read and study the reference to understand how the competency will be evaluated. (**Software Engineering Competency Model (SWECOM), IEEE Computer Society, 2014**)

ASSISTANT'S ROLES

1. One assistant is dedicated to 2-3 teams.
2. Assistant allocates a day to supervise student's team meeting once a week on a specific schedule. Assistant is also required to attend specific class session.
3. Assistant should have good communication skills.
4. Assistant should understand the competences list defined by faculty to the partner.
5. Assistant evaluates student submission for each assignment and supervises the whole project work.
6. Assistant reminds the students to prepare the evidence of their weekly progress and notes.

COMPETENCY WEEKLY CHECKLIST FORM (Example)

Main Competences	Description or Sub-Competences	Type	Final Achievement	Week						
				5	7	9	11	13	15	
Part A: Knowledge and Skills				(the highest score)						
Project Management & Lifecycle	Version control	Mandatory	0	0	0	0	0	0	0	
	Management Tools	Elective	0	0	0	0	0	0	0	
UI/UX Design	Design Review: Provide Constructive Feedbacks	Mandatory	0	0	0	0	0	0	0	
	Wireframe	Mandatory	0	0	0	0	0	0	0	
Software Design	Tool: <Example: Balsamiq>	Elective	0	0	0	0	0	0	0	
	DBMS: <Example: Mysql, MonggoDB>	Elective	0	0	0	0	0	0	0	
	Refactoring	Mandatory	0	0	0	0	0	0	0	
	Webservices, API Documentation in Apiary.io	Mandatory	0	0	0	0	0	0	0	
			Elective	0	0	0	0	0	0	0
		Advanced Design: <Example: Concurrency, fault tolerance>								
Development and Deployment	Analyse design and evaluate alternatives	Mandatory	0	0	0	0	0	0	0	
	Continuous Integration, Build Script, Continuous Deployment	Mandatory	0	0	0	0	0	0	0	
	Standard Code Convention	Mandatory	0	0	0	0	0	0	0	
	Applying suitable algorithm and data structures	Mandatory	0	0	0	0	0	0	0	
	Build Script and software Integration	Mandatory	0	0	0	0	0	0	0	
Software Quality Assurance	Unit Test and Test Coverage	Mandatory	0	0	0	0	0	0	0	
		Elective	0	0	0	0	0	0	0	
	Other Testing: <Example: Acceptance, Stress, Penetration>									
	Quality Dashboard: <Example: SonarQube>	Elective	0	0	0	0	0	0	0	
Software Sustainment & Configuration Management	Bug/Ticket Tracking	Mandatory	0	0	0	0	0	0	0	
		Elective	0	0	0	0	0	0	0	
	Software Transition <Example: Operational environment>									
Security, Privacy	Software Support <Example: Collect Operational Data>	Elective	0	0	0	0	0	0	0	
	Configuration <example: Manage Software Release>	Elective	0	0	0	0	0	0	0	
	Role Management	Mandatory	0	0	0	0	0	0	0	
	Authentication Framework: <Example: OAuth>	Elective	0	0	0	0	0	0	0	
Documentation	Secure Coding: <Example: Buffer Overflow>	Elective	0	0	0	0	0	0	0	
	Secure Deployment and Administration	Mandatory	0	0	0	0	0	0	0	
	Living Documentation	Mandatory	0	0	0	0	0	0	0	
Scalability	Documentation Generator Tool: <Example: Sphinx>	Elective	0	0	0	0	0	0	0	
	User or Installation or Deployment Manual	Elective	0	0	0	0	0	0	0	
	Software Profiling	Mandatory	0	0	0	0	0	0	0	
	Concurrency or Cloud Deployment	Elective	0	0	0	0	0	0	0	

Part B: Behavioural Attributes			(average score)						
Aptitude	Exhibited by the ability to effectively perform a software engineering task. Aptitude is not the same as knowledge or skill but rather indicates the ability (either intuitive or learned) to apply knowledge in a skillful way.	Mandatory	0	0	0	0	0	0	0
Initiative in Team work	Exhibited by: enthusiastically starting and following through on a software engineering work task; expressing and communicating interest in performing a work task; undertaking a task when asked and capably performing it, even if it is a task the individual is not enthusiastic about performing; enthusiastically and willingly with other team members while collaborating on shared tasks	Mandatory	0	0	0	0	0	0	0
Commitment	Demonstrated over time by exhibiting ethical behavior, honesty, integrity, and dependability in an individual's decisions and actions.	Mandatory	0	0	0	0	0	0	0
Work Ethic	Exhibited by being always on-time, discipline, reliable, acquiring new skills, and being willing to perform work tasks.	Mandatory	0	0	0	0	0	0	0
Cultural Sensitivity	Exhibited by an awareness of and accommodation for differences in communication styles, social interactions, dress codes, and overall behavior based on ethnic, religious, gender orientation, and other behavioral characteristics; effectively communicating a vision, strategy, method, or technique that is then accepted and shared by team members, managers, project stakeholders, and others.	Mandatory	0	0	0	0	0	0	0
Communication	Exhibited by expressing concepts, techniques, thoughts, and ideas in both oral and written forms in a clear and concise manner while interacting with team members, managers, project stakeholders, and others; includes effective listening.	Mandatory	0	0	0	0	0	0	0