

Macao Polytechnic Institute
School of Applied Sciences
Bachelor of Science in Computing
Module Outline

Academic Year 2020/2021 **Semester** 2

Learning Module	Software Engineering		Class Code	COMP223	
Pre-requisite(s)	Nil				
Medium of Instruction	English			Credit	3
Lecture Hours	45 hrs	Lab/Practice Hours	0 hrs	Total Hours	45 hrs
Instructor	Dr. Amang Kim		E-mail	amangkim.mpi@gmail.com	
Office	Rm# A320		Telephone	8599-6455	

DESCRIPTION

This course introduces the concepts of software development. Emphasis will be put on understanding the processes, techniques and methods used to develop application software. Besides, students are exposed to various software development approaches. Upon completion, students will be able to understand the major software development methodologies and techniques, appreciate their relative merits and their limitations.

LEARNING OUTCOMES

After completing the module, students will be able to:

1. Classify the problems typically encountered in each aspect of software engineering and apply alternative processes, tools and technologies for overcoming those problems (SM3p, EA2p, D1p, EP2p).
2. Identify the various types of work activities that occur and the types of work products that are generated in each phase of the software lifecycle (EA1p, D3p, ET1p, EP1p).
3. Apply various software engineering process models, and identify the strengths and weaknesses of each one (D2p, ET2p, EP5p).
4. Make use of the software design processes in order to deliver a product meeting the requirements of users (D6p, ET4p, ET3p, EP8p).

5. Illustrate software development standards and processes including those for requirements, design, construction, testing, project management and quality assurance (EA3p, D5p, ET5p, ET6p, EP4p).
6. Plan software process improvement models and the importance to the profession of applying and improving one's software engineering competencies and practices (EA4p, D4p, EP7p, EP9p).

CONTENTS:

- | | |
|---|------------------|
| 1. 1. Introduction to software engineering & process | 6.0 hours |
| 1.1 Professional software development | |
| 1.2 Software process models (2.1) | |
| 1.3 Process activities (2.2 & 2.3) | |
| 1.4 Process improvement (2.4) | |
| 2. Requirements engineering (Ch. 4) | 6.0 hours |
| 2.1 Functional and non-functional requirements | |
| 2.2 The software requirements document | |
| 2.3 Requirements specification | |
| 2.4 Requirements engineering processes | |
| 2.5 Requirements elicitation and analysis | |
| 2.6 Requirements validation | |
| 2.7 Requirements management | |
| 3. System modeling (Ch. 5) | 4.5 hours |
| 3.1 Context models | |
| 3.2 Interaction models | |
| 3.3 Structural models | |
| 3.4 Behavioral models | |
| 3.5 Model-driven engineering | |
| 4. Architectural design (Ch. 6) | 4.5 hours |
| 4.1 Architectural design decisions | |
| 4.2 Architectural views | |
| 4.3 Architectural patterns | |
| 4.4 Application architectures | |
| 5. Design and implementation (Ch. 7) | 1.5 hours |

5.1	Object-oriented design using UML	
5.2	Design patterns	
5.3	Implementation issues	
5.4	Open source development	
6.	Software Evolution (Ch. 9)	3.0 hours
6.1	Evolution processes	
6.2	Program evolution dynamics	
6.3	Software maintenance	
6.4	Legacy system management	
7.	Agile software development (Ch. 3)	3.0 hours
7.1	Agile method & Capability Maturity Model (CMM / CMMI)	
7.2	Agile development techniques	
7.3	Agile project management	
7.4	Continuous Development (CD): Beyond Agile process	
8.	Project planning (Ch. 23)	3.0 hours
8.1	Software pricing	
8.2	Plan-driven development	
8.3	Project scheduling	
8.4	Agile planning	
8.5	Estimation technique	
9.	Software Development Practice	6.0 hours
9.1	Software development in Internet era	
9.2	Case study: Com2uS mobile game development	
9.3	Group Activity: SW development startups	
10.	Quality management (Ch. 24)	3.0 hours
10.1	Software quality	
10.2	Software standards	
10.3	Review and inspections	
10.4	Software measurement and metrics	

TEACHING METHOD

Lectures, case method teaching and online (Facebook) support

ATTENDANCE

Attendance requirements are governed by the “Academic Regulations Governing Bachelor’s Degree Programmes of Macao Polytechnic Institute”. Students who do not meet the attendance requirements for the course will not be permitted to sit the final or re-sit examination and shall be awarded an ‘F’ grade.

ASSESSMENT

This learning module is graded on a 100 point scale, with 100 being the highest possible score and 50 being the passing score.

Item	Description	AHEP3 LO	Percentage
1. Popup Quiz	Home/Class-based exercises	EA1p, EA3p, ET2p, EP2p	5%
2. Take-home assignments	Home based exercises	SM3p, EA2p, ET3p, EP5p, EP4p	15%
3. Group Project	Presentation and Report	D4p, D6p, ET2p, ET4p	15%
4. Group Activity	Project management practice	D1p, D3p, ET6p, EP7p, EP8p	10%
5. Test	1.5-hour classroom test	EA4p, D2p, ET1p, EP1p	15%
6. Examination	3-hour classroom examination	EA4p, D2p, D5p, ET5p, EP9p	40%
Total Percentage:			100%

Students with an overall score of less than 35 in the coursework must take the re-sit examination even if the overall score for the module is 50 or above.

Students with a score of less than 35 in the final examination must take the re-sit examination even

if the overall score for the module is 50 or above.

Students with an overall final grade of less than 35 are NOT allowed to take the re-sit examination.

TEACHING MATERIAL:

Textbook(s)

- 1 Ian Sommerville (2015), *Software Engineering*, 10th Edition, Pearson, Boston, MA.

REFERENCE:

ARTICLES

1. S.-K. Kim (2014), **Com2uS: mobile game pioneer**, *International Journal of Teaching and Case Studies* **5:2**, pp. 189-196.
2. S.-K. Kim (2014), **Com2uS Mobile Game Development**, *Journal of Information Technology Case and Application Research* **16:3-4**, pp. 155-167.
3. M. Iansiti and A. MacCormack (1997), **Developing Products on Internet Time**, *Harvard Business Review*, **Sep.-Oct Issue**, pp. 108-117 [Access] <https://hbr.org/1997/09/developing-products-on-internet-time>
4. S.-K. Kim (2019), **Systematic Innovation Mounted Software Development Process and Intuitive Project Management Framework for Lean Startups**, *ArXiv.org* [Access] <https://arxiv.org/abs/1708.06900>
5. M. Shahin, M. Ali Babar and L. Zhu (2017), **Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices**, *IEEE Access* **5**, pp. 3909-3943.

Cases

1. S.-K. Kim (2019), **Developing Product on Internet Time: A Process Design Exercise (Revised)**, *MPI Case Bank*, 3 pages
2. J. J. Bussgang and et al. (2018), **Continuous Software Development: Agile's Successor**, *Harvard Business School* **9-818-050**, 17 pages.
3. R. D. Austin (2008), **CMM versus Agile: Methodology Wars in Software Development**, *Harvard Business School* **9-607-084**, 17 pages.