King Saud University
College of Computer and Information Sciences
Department of Software Engineering

Course Code: SWE 313
Title: Software Process and Modeling
Credit Hours: 3 (3-0-1)
Semester: Fall 2016, Academic Year 2016/2017

Course Syllabus

1. Course Identification and General Information:

Course Title: Software Process and Modeling	**Course Code: SWE 313**
Course Level: 6	**Course Prerequisite: SWE 211– Introduction to Software Engineering**
Lecture Time: 3 hours lecture and 1 hour tutorial/lab per week	**Credit Hours: 3 (3-0-1)**

2. Course Description:

This course aims to provide a comprehensive presentation of the key concepts, modeling techniques, and development methodologies used in object-oriented approaches in software engineering. This includes a review of Object-Oriented concepts and modeling with UML: Structural Modeling, Behavioral Modeling – System architecture design, – User Interface Design – Object Persistence Design - Class and Method Design - Object-Oriented Testing – Unified Process development cycle – Use case analysis – Sequence diagrams – Encapsulation – Inheritance – Polymorphism – Design principles of coupling and cohesion – Design patterns. Students will be also exposed OO case tools, UML Generating tools, standard templates, Quality control and other SWE related standards. Students will participate in a group project on object-oriented software methodologies and modeling using OO case tools.

3. Course Objectives:

1. Define and understand fundamental and advanced concepts of software engineering process.
2. Define fundamental and advanced OO Software Engineering concepts.
3. Understand how to capture system requirements in use cases.
4. Understand how to transform an analysis models into to design models.
5. Apply an iterative process to the development of a design model.
6. Design UML diagrams to represent analysis and design models.
7. Understand software processes and software development methodologies.
8. Use OO Case tools to create UML diagrams.

4. Instructor:

- Dr. Mohammed F. Alhamid
- Address: Building No. 31, Office No. 2116 | Phone: 0114673525 . **You will find me most of the time at the Department Chair Office.**
- Email: mohalhamid@ksu.edu.sa
- Website: http://fac.ksu.edu.sa/mohalhamid
- Office hours: TBD.

5. Relationship of Course to Program Outcomes:

Outcome	Outcome Description	Contribution
(c)	An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability	**M**
(e)	An ability to identify, formulate, and solve engineering problems	**M**
(k)	An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice	**H**
(l)	The ability to analyze, design, verify, validate, implement, apply, and maintain software systems	**H**

H=High, **M**= Medium, **L**=Low

6. Course References:

6.1 Textbooks:

Primary:

- Object-oriented Software Engineering: Practical Software Development, by Robert Laganière and Timothy C. Lethbridge.

Supplementary:

- Software Modeling and Design (2011) by Hassan Gomaa, Cambridge university press.

6.2 Support Material (Journals, Publications, etc):

https://lms.ksu.edu.sa

6.3 Study Guide(s) (if applicable):

https://lms.ksu.edu.sa

6.4 Homework and Laboratory Guide(s) (if applicable):

https://lms.ksu.edu.sa

7. Teaching Methods:

Lectures, Discussion Groups using practical examples, Tutorials, etc.

8. Learning Outcomes:

8.1 Knowledge and Understanding:

The student will gain knowledge and understanding of:

1. Understanding the concepts of software process
2. Define fundamental and advanced OO Software Engineering concepts
3. Understand how to capture system requirements in use cases
4. Understand how to transform an analysis models into to design models
5. Apply an iterative process to the development of a design model.
6. Describe some basic design considerations, including the use of design patterns.
7. Use of different UML Diagrams to represent analysis and design models.
8. Use the techniques of forward and reverse engineering to generate code from UML models and *vice-versa*.
9. Understand Software Processes and Software development methodologies (such as RUP).
10. USE OO Case tools (such as IBM Rational Rose) to create UML diagrams.

8.2 Cognitive Skills (Thinking and Analysis):

The students will acquire the necessary skills to apply the different activities of the software engineering process in developing actual web applications. The focus will be on the first four activities: communication, planning, analysis, and design.

9. Methods of Assessment:

Assessment Instruments	Mark
Midterm ExaSmination	20
Quizzes (2 x 2.5)	5
Assignments (2 x 5)	10
Project (presentation 10%, report 10%) • Presentation can be substituted with a tutorial video.	20
Labs	5
Final Examination	40
Total	**100**

10. Course Policies:

- No late assignment will be accepted.
- The quizzes may be given at anytime during class-time.
- An inclusive final exam will be given.

11. Course Academic Calendar

Please be adviced this is not a final table of contents, I might still add/modify some chapters as we progress in the course.

# of Week	# of hours	Basic and support material to be covered	Homework/reports and their due dates	expected
(0.5)	2	**Introduction and Terminologies:** nature of software, activities common to Software Engineering, Modeling, Process.		
(1)	4	**Software Process:** process models, the opportunistic approach, waterfall model, spiral model, concurrent engineering model, The Rational Unified Process (RUP), Agile approaches, XP, COTS, Reengineering.		
(1)	4	**Modeling Principles:** modeling, metamodeling, model transformation, types of models, tools support, Model Driven Architecture (MDA), modeling levels, modeling languages, DSL vs. GPL.		
(2)	8	**Modeling with UML:** what why and how, UML diagrams, use case diagrams (use case extensions, generalization, inclusion, class diagrams (classes, relationships, associations and multiplicity, aggregation and composition), sequence diagrams (interactions, life spans), state diagrams, UML modeling tools.		
(2)	8	**Focus on Modeling Interaction and behaviors:** interaction diagrams, iterations and sequence, communication diagram, how to use communication vs. sequence diagrams, state diagrams (transitions and activities), activity diagrams, concurrency,		
(2)	8	**Modeling real-time & concurrent applications Case Study:** COMET modeling methodology, static and dynamic modeling, distributed components, real world case study,		
(2)	8	**Metamodeling and languages:** Extensibility and stereotype, multiple stereotyping, UML profiles, Profile Examples (MARTE, SysML, …), domain-specific languages (classification, DSL examples).		
(1)	3	Final Exam		

12. Expected Workload:

On average each student should spend 4 hours per week studying the material of the course as well as working on assignments/project.

13. Attendance Policy:

Absence from lectures and/or tutorials shall not exceed 25%. Students who exceed the 25% limit without an accepted medical or emergency excuse shall not be allowed to take the final examination and shall receive a grade of "DN" for the course.