MODULE DESCRIPTION FORM

Module Information			
Module Title	Software Engineering		
Module Type	Core		
Module Code	SWE301		
ECTS Credits	6		
SWL (hr/sem)	150		
Module Level	UG 3	**Semester**	5
Administering Department	Computer Science	**College**	Collage of Science
Module Leader	Dr. Abdulbasit	**dr.abdulbasit@uodiyala.edu.iq**	

Relation with other Modules		
Prerequisite module	None	**Semester**
Co-requisites module	None	**Semester**

Module Aims, Learning Outcomes and Indicative Contents

Module Objectives

This module aims to provide students with a comprehensive understanding of the principles, methods, and tools used in professional software engineering, following the framework of Roger S. Pressman's "Software Engineering: A Practitioner's Approach". Key objectives include:

- Understanding the foundational concepts of software engineering and the software process.
- Mastering various software process models (Waterfall, Incremental, Evolutionary, Agile).
- Learning to elicit, analyze, and specify software requirements.
- Developing skills in software design (architectural, component-level, UI/UX, mobility).
- Implementing rigorous quality assurance, security engineering, and testing strategies.
- Applying project management concepts (planning, risk management, metrics).
- Exploring advanced topics (process improvement, emerging trends).

Module Learning Outcomes

Upon successful completion of this module, students will be able to:

1. Select appropriate software process models for a given project.
2. Elicit, document, and validate software requirements.
3. Design and model software architectures using UML and design patterns.
4. Develop and implement test plans at component, integration, and system levels.
5. Apply software quality assurance and security engineering principles.
6. Manage a small-scale software project using basic project management techniques.
7. Critically evaluate emerging trends in software engineering.

Indicative Contents

The content follows Pressman's 9th edition:

Part One: The Software Process – Ch.1-5

Part Two: Modeling – Ch.6-14 (Requirements, Design, UX, Mobility)

Part Three: Quality and Security – Ch.15-23 (Quality, Security, Testing, Configuration Management)

Part Four: Managing Software Projects – Ch.24-27 (Project Management, Risk, Support)

Part Five: Advanced Topics – Ch.28-30 (Process Improvement, Emerging Trends)

Learning and Teaching Strategies
The module is delivered through a balanced combination of theoretical lectures and practical laboratory sessions. Lectures introduce core concepts, theories, and methodologies, supported by real-world case studies. Laboratory sessions provide hands-on experience with modern software engineering tools (e.g., Git, UML modeling tools, CI/CD pipelines). Students work in small teams on a semester-long project, simulating a real-world development environment.

Student Workload (SWL)			
Structured SWL (h/sem)	64	**Structured SWL (h/w)**	4
Unstructured SWL (h/sem)	68	**Unstructured SWL (h/w)**	4
Total SWL (h/sem)		150	

Module Evaluation				
As	**Time/Number**	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative	Quizzes (2)	10% (10)	2,7	1,2
	Assignments (2)	10% (10)	2,10	1,3,5
	Project/Lab	10% (10)	Continuous	2,3,4,6
	Report (1)	10% (10)	13	6,7
Summative	Midterm Exam (2hr)	10% (10)	7	1,2,3,4
	Final Exam (3hr)	50% (50)	16	All
Total assessment		100% (100 Marks)		

Delivery Plan (Weekly Syllabus)	
Week	**Material Covered**
1	Introduction to Software Engineering; The Software Process (Ch.1)
2	Process Models: Waterfall, Incremental, Evolutionary (Ch.2)
3	Agility and Agile Methodologies (Ch.3)
4	Human Aspects of Software Engineering (Ch.5)
5	Understanding Requirements (Ch.7)
6	Requirements Modeling (Ch.8)
7	**Midterm Exam**
8	Design Concepts (Ch.9)
9	Architectural and Component-Level Design (Ch.10-11)
10	User Experience and Mobility Design (Ch.12-13)
11	Quality Concepts and Software Quality Assurance (Ch.15-17)
12	Software Security Engineering (Ch.18)
13	Software Testing Strategies (Ch.19-21)
14	Project Management and Risk Management (Ch.24-27)
15	Advanced Topics: Process Improvement, Emerging Trends (Ch.28-30)
16	**Final Exam**

Delivery Plan (Weekly Lab. Syllabus)	
Week	**Lab Activity**
1	Introduction to UML: Use Case and Class Diagrams
2	Requirements Engineering: Writing a Specification
3	Agile with Scrum/Kanban
4	Version Control: Git and GitHub
5	Requirements Modeling: Activity and Sequence Diagrams
6	Software Design: Architectural Patterns
7	Design Review and Refactoring
8	Unit Testing with JUnit/PyTest; TDD
9	Continuous Integration with GitHub Actions
10	Code Reviews and Static Analysis
11	Project Management with Jira/Trello
12	Final Project: Implementation and Integration
13	Final Project: System Testing
14	Final Project: Demo Preparation
15	Final Project: Live Demonstrations
16	No Lab

Learning and Teaching Resources		
	Text	**Available in Library?**
Required Texts	Pressman, R. S., & Maxim, B. R. (2024). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill.	Yes
Recommended Texts	Sommerville, I. (2019). *Software Engineering* (10th ed.). Pearson.	Yes
Websites	`https://www.omg.org/uml/`, `https://www.atlassian.com/agile`	N/A

Grading Scheme				
Group	**Grade**	**Arabic Equivalent**	**Marks %**	**Definition**
Success (50-100)	A - Excellent	Excellent	90-100	Outstanding Performance
	B - Very Good	Very Good	80-89	Above average with some errors
	C - Good	Good	70-79	Sound work with notable errors
	D - Satisfactory	Satisfactory	60-69	Fair but with major shortcomings
	E - Sufficient	Sufficient	50-59	Work meets minimum criteria
Fail (0-49)	FX - Fail	Fail (under review)	45-49	More work required but credit awarded
	F - Fail	Fail	0-44	Considerable amount of work required
Note: Marks decimal places above or below 0.5 will be rounded to the higher or lower full mark. The University has a policy NOT to condone "near-pass fails".				