Course Specification

(Bachelor)

Course Title:	Software Maintenance and Evolution
Course Code:	SE4704
Program:	BSc in Software Engineering
Department:	Software Engineering
College:	College of Computing
Institution:	Umm Al Qura University
Version:	1.0
Last Revision Date:	22/04/2025

Table of Contents

A. General information about the course:

1. Course Identification

1. Credit hours: (3)					
2. Course type					
A.	☐University	☐College	☒ Department	☐Track	☐Others
B.	☐ Required		☒ Elective		
3. Level/year at which this course is offered: (3rd year/ 5th or 6th level) or (4th year/8th level)					
4. Course General Description:					
This course emphasizes the principles, practices, and challenges of maintaining and evolving software systems to ensure their long-term usability and value. It covers maintenance models, reengineering techniques, impact analysis, refactoring, and program comprehension. Students will learn how to address issues related to managing legacy systems, reduce maintenance costs, and improve software quality through reuse and domain engineering.					
5. Pre-requirements for this course (if any)**:**					
SE3103 - Software Testing and Quality Assurance					
6. Co-requisites for this course (if any)**:**					
N/A					
7. Course Main Objective(s):					
Upon successful completion of this unit, students should be able to: 1. Understand the principles and challenges of software maintenance and evolution, including the management of legacy systems and their impact on software sustainability and reliability**.** 2. Analyze and apply appropriate models, techniques, and strategies for software reengineering, refactoring, and program comprehension to improve system quality and maintainability. 3. Develop skills to assess the impact of software changes through impact analysis and estimate maintenance costs to ensure efficient resource utilization. 4. Promote ethical and professional practices in software maintenance, emphasizing collaborative teamwork, regulatory compliance, and societal considerations.					

2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	45	100%
2	E-learning	0	0
3	Hybrid • Traditional classroom • E-learning	0	0
4	Distance learning	0	0

3. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	**Lectures**	45
2.	**Laboratory/Studio**	0
3.	**Field**	0
4.	**Tutorial**	0
5.	**Others (specify)**	0
Total		45

B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
1.0	**Knowledge and understanding**			
1.1	Explain the principles and challenges of software maintenance and evolution, including types, models, and the management of legacy systems.	K1	Lectures, Interactive Discussions	Quizzes, Assignments, Exams
1.2	Evaluate the importance of software reuse and domain engineering in reducing	K2	Lectures, Interactive Discussions	Quizzes, Assignments, Exams

Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
	maintenance costs and improving system efficiency.			
2.0	**Skills**			
2.1	Analyze and apply models, techniques, and strategies for software reengineering, refactoring, and program comprehension to improve system quality.	S1	Problem-solving exercises, Group projects	Assignments, Project reports and presentations
2.2	Apply refactoring techniques to enhance code maintainability while adhering to software quality standards.	S2	Practical coding exercises, Code review sessions	Code review evaluation assignment, Project reports and presentations
2.3	Utilize frameworks to support software reuse and modularity	S3	Exercises, Group projects	Assignments, Project reports and presentations
3.0	**Values, autonomy, and responsibility**			
3.1	Demonstrate ethical standards and professional responsibilities in software maintenance and evolution, emphasizing sustainability, fairness, and societal impact.	V1	Group Discussions, Guest Lectures	Reflection Essays on Ethical Scenarios.
3.2	Collaborate within a team to effectively address maintenance challenges and implement evolution strategies that align with industry standards and best practices.	V2	Group Projects, Tutorials	Group Project

C. Course Content

No	List of Topics	Contact Hours
1.	Introduction to Software Maintenance and Evolution	**3**

2.	Maintenance and Evolution Types	3
3.	Maintenance and Evolution Models	3
4.	Software Reengineering	3
5.	Managing Legacy Information Systems	6
6.	Impact Analysis	6
7.	Refactoring and Program Comprehension Techniques	9
8.	Maintenance Cost Estimation and Optimization	6
9.	AI-Assisted Code Evolution and Predictive	6
Total		45

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Quizzes	2-14	15
2.	Projects	2-14	15
4.	Assignments	2-14	10
5.	Mid Term	7	20
6.	Final Exam	16-17	40

*Assessment Activities (i.e., Written test, oral test, oral presentation, group project, essay, etc.).

E. Learning Resources and Facilities

1. References and Learning Resources

Essential References	• Tripathy, P., & Naik, K. (2014). *Software evolution and maintenance: A practitioner's approach*. Wiley & Sons, Inc. ISBN 978-0470603413. (wiley.com) • Sommerville, I. (2015). *Software engineering* (10th ed.). Pearson. ISBN 978-0133943030. (pearson.com)
Supportive References	• Winters, T., Manshreck, T., & Wright, H. (2020). *Software engineering at Google: Lessons learned from programming over time*. O'Reilly Media. ISBN 978-1492082798.
Electronic Materials	• IEEE Computer Society. (n.d.). *Software maintenance course*. Retrieved from https://www.computer.org/product/education/software-maintenance-course
Other Learning Materials	

2. Required Facilities and equipment

Items	Resources
facilities (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	Classroom
Technology equipment (projector, smart board, software)	Projector
Other equipment (depending on the nature of the specialty)	N/A

F. Assessment of Course Quality

Assessment Areas/Issues	Assessor	Assessment Methods
Effectiveness of teaching	Students	Direct, Indirect
Effectiveness of Students' assessment	Faculty, Peer reviewer	Direct, Indirect
Quality of learning resources	Faculty, Course coordinator	Direct, Indirect
The extent to which CLOs have been achieved	Course coordinator, Program management committee	Direct
Other		

Assessors (Students, Faculty, Program Leaders, Peer Reviewers, Others (specify))

Assessment Methods (Direct, Indirect)

G. Specification Approval

COUNCIL /COMMITTEE	**SOFTWARE ENGINEERING DEPARTMENT COUNCIL**
REFERENCE NO.	**THE 17TH MEETING FOR THE ACADEMIC YEAR 1446H**
DATE	**22/04/2025**