



Course Specification

(Bachelor)

Course Title: **Sustainability of Software Engineering**

Course Code: **SE4706**

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:	3
B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods	4
C. Course Content	5
D. Students Assessment Activities	6
E. Learning Resources and Facilities	6
F. Assessment of Course Quality	7
G. Specification Approval	7



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:

The "Sustainability of Software Engineering" course examines the principles, practices, and tools involved in creating and maintaining sustainable software systems. It focuses on integrating environmental, social, technical and economic sustainability considerations throughout all stages of the software development lifecycle. Students will learn about sustainability theories, methodologies, and tools, while also developing the skills necessary to evaluate and incorporate sustainable practices into software engineering processes, architecture, and testing.

5. Pre-requirements for this course (if any):

SE3102 - Software Design and Architecture

6. Co-requisites for this course (if any):

N/A

7. Course Main Objective(s):

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

1. Acquire a comprehensive understanding of sustainability concepts and their relevance to software engineering.
2. Develop skills to analyze, model, and design software systems with sustainability as a fundamental principle.
3. Cultivate the ability to integrate sustainability into software engineering processes, including requirements gathering, architecture, testing, and quality assurance.
4. Encourage the application of sustainable practices using appropriate tools, methods, and technologies in real-world software projects.

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	0	0





No	Mode of Instruction	Contact Hours	Percentage
	<ul style="list-style-type: none"> Traditional classroom E-learning 		
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	Understand and explain the key concepts and principles of sustainable software engineering.	K1	Lectures, Case studies	Quizzes, Assignments, Exam
1.2	Develop sustainable testing strategies and quality assurance processes aligned with eco-design standards.	K1	Lectures, Case studies	Quizzes, Assignments, Exam
2.0	Skills			
2.1	Analyze and evaluate sustainability issues in software requirements, architecture, and development processes.	S1	Discussions, Case study analysis	Exercises, Group Project
2.2	Apply sustainability principles in designing and developing software systems, ensuring energy efficiency and resource optimization.	S3	Hands-on lab sessions, Tool demonstrations	Exercises, Group Project
2.3	Model and address sustainability conflicts using	S2	Modeling workshops with	Exercises, Group Project



Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
	UML and other tools to create sustainable system designs.		sustainability scenarios	
3.0	Values, autonomy, and responsibility			
3.1	Demonstrate awareness of professional and ethical responsibilities in fostering sustainability in software engineering.	V1	Discussions on ethical case studies	Reflective essays
3.2	Collaborate on capstone projects to propose innovative, sustainable software solutions.	V2	Collaborative brainstorming sessions, Group projects	Group Project , presentations

C. Course Content

No	List of Topics	Contact Hours
1.	Introduction to Sustainable Software Engineering <ul style="list-style-type: none"> Overview of sustainability concepts in software engineering. Importance of green software in development, deployment, and operations. 	6
2.	Sustainability in Requirements Engineering <ul style="list-style-type: none"> Integrating sustainability into stakeholder, goal, process, and system modeling. Identifying and analyzing sustainability goals during requirements elicitation. 	6
3.	Sustainability Analysis and Impact <ul style="list-style-type: none"> Assessing sustainability from cost estimation to long-term environmental and social impacts. Exploring sustainability in project management, maintenance, and evolution. 	6
4.	Sustainability in Software Architecture and Design <ul style="list-style-type: none"> Incorporating sustainability in database design, human-computer interaction, and modular systems. Principles of eco-friendly software architecture development. 	6
5.	Sustainable Engineering Processes <ul style="list-style-type: none"> Improving software processes with a focus on energy/resource efficiency. Agile and other methodologies for sustainable software engineering. 	6
6.	Sustainability in Testing and Quality Assurance	6





	<ul style="list-style-type: none"> Developing eco-friendly testing strategies and aligning them with ISO eco-design standards. Approaches to ensuring long-term software quality. 	
7.	Sustainability Management and Policies <ul style="list-style-type: none"> Planning and controlling sustainability activities. Formulating sustainability policies and eco-design for digital services. 	3
8.	Tools for Sustainability in Software Development <ul style="list-style-type: none"> Tools for requirements, design, testing, and configuration management. Creating a laboratory program for green software engineering practices. 	3
9.	Advanced Applications <ul style="list-style-type: none"> Addressing sustainability in parallel computing, clusters, and grids. Techniques for maintaining sustainability in high-performance systems. 	3
Total		45

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Quizzes	4,12	15
2.	Project	12	15
3.	Assignments	2-14	10
4.	Mid Term	8	20
5.	Final Exam	16	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

- Currie, A., Hsu, S., & Bergman, S. (2024). *Building green software: A sustainable approach to software development and operations*. O'Reilly Media. ISBN 978-1098150624.
- Lilienthal, C. (2019). *Sustainable software architecture: Analyze and reduce technical debt*. Rocky Nook. ISBN 978-1681985695.





Supportive References	<ul style="list-style-type: none"> Fontanarrosa, S. (2024). <i>Green software engineering: Exploring green technology for sustainable IT solutions</i>. Packt Publishing. ISBN 978-1835885888. Calero, C., & Piattini, M. (Eds.). (2015). <i>Green in software engineering</i>. Springer. ISBN 978-3319085807.
Electronic Materials	
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

