Course Specification

(Bachelor)

Course Title:	Software Metrics
Course Code:	SE4205
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: (2)					
2. Course type					
A.	☐University	☐College	☒ Department	☐Track	☐Others
B.	☒ Required		☐ Elective		
3. Level/year at which this course is offered: (4th year/ 8th level)					
4. Course General Description:					
This course uncovers what software metrics are, why they are used, who should develop the metrics, when they should be applied, who should own the measurements, and what should be done with them. It explores the current definitions of useful metrics related to software engineering. It provides (i) the theoretical definition of measurement and its application to software engineering; (ii) a comprehensive review of the state of the art in measurement for software engineering; (iii) guidelines on establishing a measurement program; (iv) cost-benefit analyses for selected industrial case studies of software measurement programs.					
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):					
This course will teach students the methods, theory and tools vital to the key principles of software systems measurement. The course will present state-of-the-practice as well as research investigations pushing the current envelope. The course aims to give students an appreciation for the challenges and solutions for measurement throughout the software lifecycle.					

2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	30	100%
2	E-learning	0	0
3	Hybrid • Traditional classroom	0	0

No	Mode of Instruction	Contact Hours	Percentage
	• E-learning		
4	Distance learning	0	0

3. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	**Lectures**	30
2.	**Laboratory/Studio**	0
3.	**Field**	0
4.	**Tutorial**	0
5.	**Others (specify)**	0
Total		30

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 the fundamental concepts of software measurement and metrics, including property-oriented measurement.	K1	Lectures, Group Discussions	Quizzes, Exams
1.2	Explore measurement frameworks like Goal-Question-Metrics (GQM) and software quality models such as ISO 9126.	K2	Lectures, Group Discussions	Quizzes, Exams
1.3	Learn about software size, structure, and complexity metrics, including cyclomatic complexity and control-flow.	K1	Lectures, Group Discussions	Quizzes, Exams
2.0	**Skills**			

Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
2.1	Apply software metrics to measure internal and external attributes such as quality, reliability, cost, and effort.	S1	Lectures, Group Discussions	Quizzes, Exams
2.2	Implement empirical investigation techniques, including formal experiments and case studies, in software engineering.	S2	Lectures, Group Discussions	Quizzes, Exams
2.3	Evaluate object-oriented and agile metrics for software productivity and process improvement.	S3	Lectures, Group Discussions	Quizzes, Exams
3.0	**Values, autonomy, and responsibility**			
3.1	Recognize the importance of measurement accuracy, validation, and meaningfulness in software metrics.	V1	Lectures, Group Discussions	Quizzes, Exams
3.2	Appreciate the role of metrics in ensuring software quality, reliability, and customer satisfaction.	V1	Lectures, Group Discussions	Quizzes, Exams
3.3	Embrace a goal-based approach to software measurement to drive continuous improvement in software development.	V1	Lectures, Group Discussions	Quizzes, Exams

C. Course Content

No	List of Topics	Contact Hours
1.	Overview of software metrics Introducing the course. What is software measurement? What are software metrics?	**2**

2.	The basics of measurement Metrology Property-oriented measurement Meaningfulness in measurement Measurement quality Measurement process Measurement Scales Measurement validation Object-oriented measurement Subject-domain-oriented measurement	**4**
3.	Goal-based framework for software measurement Software measure classification Goal-based paradigms: Goal-Question-Metrics (GQM) and others TBD Applications of GQM Case studies	**2**
4.	Empirical investigation Software engineering investigation Investigation principles Investigation techniques Formal experiments overview: Planning, Principles, Types and Selection Guidelines for empirical research	**2**
5.	Measuring internal product attributes: size Software size Software Size: Length (code, specification, design) Software Size: Reuse Software Size: Functionality (function point, feature point, object point, use-case point) Software Size: Complexity	**2**
6.	Measuring internal product attributes: structure Software structural measurement Control-flow structure Cyclomatic complexity Data flow and data structure attributes Architectural measurement	**2**
7.	Measuring cost and effort Software cost models COCOMO and COCOMO II Constraint model Software Lifecycle Management (SLIM)	**2**

8.	Measuring external product attributes: quality Software quality Software quality models: Boehm's model, McCall's model, ISO 9126 model, etc. Basic software quality metrics Quality management models Measuring customer satisfaction Software Quality Assurance (SQA)	**2**
9.	Measuring software reliability Reliability concepts and definitions Software reliability models and metrics Fundamentals of software reliability engineering (SRE) Reliability management models	**2**
10.	Software test metrics Test concepts, definitions and techniques Estimating number of test cases Allocating test times Decisions based on testing Test coverage measurement Software testability measurement Remaining defects measurement	**2**
11.	Object-oriented metrics Object-Oriented measurement concepts Basic metrics for OO systems OO analysis and design metrics Metrics for productivity measurement Metrics for OO software quality Experience-based guidelines	**2**
12.	Agile metrics Extreme Programming metrics Scrum metrics Agile RUP, Others	**2**
13.	Cost of Quality/Ownership, ROI metrics, Organizational metrics Extended lifecycle cost and process maturity metrics e.g. CoSQ, COO, CMMI	**2**
14.	SWE Lifecycle Model Considerations	**2**
Total		**30**

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Assignments and Quizzes	2-14	15
2.	Projects	2-14	15
3.	Practicals	2-14	10
4.	Mid Term	7	20
5.	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	• Fenton, N., & Bieman, J. (2014). *Software metrics: A rigorous and practical approach* (3rd ed.). CRC Press. ISBN 978-1439838228.
Supportive References	• Fenton, N., & Bieman, J. (2014). *Software metrics: A rigorous and practical approach* (3rd ed.). CRC Press. ISBN 978-1439838228.
Electronic Materials	
Other Learning Materials	

2. Required Facilities and equipment

Items	Resources
facilities (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	Traditional Classroom
Technology equipment (projector, smart board, software)	Multimedia 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

Assessment Areas/Issues	Assessor	Assessment Methods
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**