MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	Software Engineering			**Module Delivery**	
Module Type	**Core**			☒**Theory** ☒**Lecture** ☐**Lab** ☐**Tutorial** ☐**Practical** ☐**Seminar**	
Module Code	**MU010602106**				
ECTS Credits	**4**				
SWL (hr/sem)	**100**				
Module Level	UGII		**Semester of Delivery**	3	
Administering Department	Computer Science		**College**	College of Science	
Module Leader	Dr. Huda Haikal Hussein		**e-mail**	huda.atmsc@uomustansiriyah.edu.iq	
Module Leader's Acad. Title	Lecturer		**Module Leader's Qualification**		
Module Tutor	Lecturer Zeyad Farooq Lutfi		**e-mail**	Zeyadfa6@uomustansiriyah.edu.iq	
Peer Reviewer Name	D. Hudi Abdulalli		**e-mail**		
Scientific Committee Approval Date	25 / 4 /2024		**Version Number**	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	None	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents

أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

Module Objectives أهداف المادة الدراسية	This course covers the fundamentals of software engineering, including system requirements, effective methods of design, and testing, teams' software development, and the application of engineering tools. Also, the course will initiate students to the different software process models, software requirements engineering process, systems analysis and design as a problem-solving activity, key elements of analysis and design, phases within the system development life.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon completion of the course, the students will be able to: 1. Explain the introduction of software engineering practices that are key components. 2. Identify methods that will lead to the creation of a software architecture that achieves a specified level of reliability, dependability and security. 3. Apply the basic principles of software project management in a team environment and introduce different application types. 4. Introduce different case studies for more explanation. 5. Understand a variety of strategies to the software process models and explain their benefits and problems. 6. Understand a variety of strategies to the testing of simple programs. 7. Identify the principal issues associated with software evolution and explain their impact on the software lifecycle. 8. Discuss the system requirement type: functional and non-functional requirements.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. **UNIT I:** [15 hrs] - Introduction to Software Engineering, Software costs, Software products, Product specification, and frequently asked questions about Software Engineering. - Essential attributes of good software, Importance of software engineering, Software process activities, Application types, Software engineering fundamentals, and Web-based software engineering. - Case studies, A personal insulin pump, A mental health case patient management system, and A wilderness weather station. **UNIT-II**: [16 hrs] - Software Processes, Software process descriptions, Plan-driven and agile processes, and Software process models. Iterative Waterfall Model, Incremental development, Reuse-oriented software engineering, Incremental development benefits, Incremental development problems. Process activities, the

	requirements engineering process, Software design and implementation, A general model of the design process, Requirements Analysis and Specification. **UNIT-III** : [12 hrs] Rapid software development, Agile methods, Agile manifesto, The principles of agile. **UNIT IIII**: [20 hrs] - Requirements Engineering, what is a requirement? Types of requirements, User and system requirements. System stakeholders, Stakeholders in the Mentcare system. Functional and non-functional requirements, Mentcare system: functional requirements, Types of nonfunctional requirement, Non-functional requirements implementation, Non-functional classifications.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	**<u>Learning Strategies:</u>** The best way to learn and explain software engineering models and discuss the differences among them with their benefits and problems. You can also find books, courses, or tutorials that offer case studies for development system. 1- software engineering models Reading and Analysis: Study and analyze existing model examples and systems written by experienced developers. Understand. 2- Collaborative Learning: Engage in discussions and collaborations with fellow learners or join programming communities and forums to share knowledge, ask questions, and solve problems collectively. **<u>Teaching Strategies:</u>** 1- Conceptual Explanation: Start by providing clear and concise explanations of software engineering concepts. Use real-life examples for case studies to help students grasp abstract concepts easily. 2- Hands-on Coding Exercises: Include assignments that reinforce the concepts taught for types of software engineering models. Give examples for using use case models to identify system requirements.

	Encourage students to apply the model into practice, helping them develop their understanding skill. 3- Interactive Sessions: Conduct live sessions where students can actively participate, ask questions, and solve problems alongside the instructor. This approach promotes engagement and practical learning. In addition to the practical project 4- Resources and References: Share relevant software engineering design and analysis resources, recommended textbooks, and online tutorials to supplement the curriculum and encourage further exploration.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	63	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	4
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	37	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	2
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	**100**		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	4	12% (12)	3, 7, 9 and 12	LO #1, #2, #3, #4, #5
	Assignments	2	10% (10)	5 and 14	LO #5, #6, #7, #8
	Seminar/	1	10% (10)	Continuous	All
	Report	1	8% (8)	Continuous	All
Summative assessment	**Midterm Exam**	1hr	10 % (10)	7	LO #1, #2, #3, #4, #5
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction to Software Engineering, Software costs, Software products, Product specification, frequently asked questions about Software Engineering.
Week 2	Essential attributes of good software, Importance of software engineering, Software process activities, Application types, Software engineering fundamentals, Web-based software engineering.
Week 3	Case studies, A personal insulin pump, A mental health case patient management system, and A wilderness weather station.
Week 4	Software Processes, Software process descriptions, Plan-driven and agile processes, and Software process models.
Week 5	Iterative Waterfall Model, Incremental development, Reuse-oriented software engineering, Incremental development benefits, Incremental development problems.
Week 6	Process activities, the requirements engineering process, Software design and implementation, A general model of the design process, Requirements Analysis and Specification.
Week 7	**Midterm Exam /**
Week 8	Software Analysis & Design Tools, Structured Design, Software evolution, System evolution.
Week 9	Rapid software development, Agile methods, Agile manifesto, The principles of agile.
Week 10	Requirements Engineering, what is a requirement? Types of requirements, User and system requirements.
Week 11	System stakeholders, Stakeholders in the Mentcare system.
Week 12	Functional and non-functional requirements, Mentcare system: functional requirements, Types of nonfunctional requirement, Non-functional requirements implementation, Non-functional classifications.
Week 13	Requirements discovery, Interviewing, Interviews in practice, Problems with interviews.
Week 14	Stories and scenarios, Scenarios, Use cases, Use cases for the Mentcare system.
Week 15	**Seminar**
Week 16	Preparatory week before the Final Exam

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	
Week 2	
Week 3	
Week 4	

Week 5	
Week 6	
Week 7	
Week 8	
Week 9	
Week 10	
Week 11	
Week 12	
Week 13	
Week 14	
Week 15	

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	**[1]: Sommerville, Ian. Software engineering / Ian Sommerville. — 9th ed & 10th. ISBN 10: 0-13-703515- 2, ISBN 13: 978-0-13-703515-1. Publisher: Copyright © 2011, 2006, 2005, 2001, 1996 Pearson Education, Inc., publishing as Addison-Wesley.**	
Recommended Texts	**[1]: Mall Rajib, Fundamentals of Software Engineering, PHI.** **[2]: Pressman, Software Engineering Practitioner's Approach, TMH.**	
Websites		

Grading Scheme مخطط الدرجات				
Group	**Grade**	التقدير	**Marks %**	**Definition**
Success Group (50 - 100)	**A -** Excellent	امتياز	90 – 100	Outstanding Performance
	B - Very Good	جيد جدا	80 – 89	Above average with some errors
	C – Good	جيد	70 – 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 – 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 – 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(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 (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.