

CO3	*	*	*	*	*	*		*	*
CO4	*	*	*					*	*

Wks Mapped with the course

WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
*	*	*	*	*	*		*	*

Mark Distribution:

Total Marks	CIE Marks	ESE Marks	ESE Duration
100	40	60	2.5 hrs

Continuous Internal Evaluation Pattern:

Attendance : 5 marks

Continuous Assessment Test : 20 marks

Continuous Assessment Assignment : 15 marks

SYLLABUS

Module 1 (12 hrs)

Introduction to Software Engineering: Software engineering, Software characteristics and types. Process activities - Software specification, Software design and implementation. Software Process models - Waterfall model, Prototype model, Spiral model, Incremental model, Agile model – Values and Principles, Agile development techniques, Extreme Programming. Case studies: An insulin pump control system, Mentcare – a patient information system for mental health care.

Module 2 (10 hrs)

Feasibility study - Feasibility analysis and its types, SRS document characteristics and its structure. Functional and non-functional requirements, Requirements engineering processes. Requirements elicitation, Requirements validation, Requirements change, Traceability Matrix. Developing use cases, Software Requirements Specification Template, Personas, Scenarios, User stories, Feature identification. Design concepts - Design within the context of software engineering, Design Process, Design concepts, Design Model. Architectural Design - Software Architecture, Architectural Styles, Architectural considerations, Architectural Design Component level design.

Module 3 (12 hrs)

Coupling and cohesion. Object-oriented software design. UML diagrams and relationships: Use case, Sequence, Class, State, and Data Flow diagrams; static and dynamic models. Software design patterns: Model-View-Controller (MVC) and creational patterns. Open-source development and licensing (GPL, LGPL, BSD). Software reviews: informal and formal technical reviews, code review, and defect impact. Software testing: unit, integration, validation, system, white-box and black-box testing, debugging, test automation, test-driven development, and security testing. Overview of DevOps and code management: version control and CI/CD. Software maintenance.

Module 4 (12 hrs)

Software Configuration Management (SCM) - objectives and importance of SCM, configuration identification and baseline management, version control and change control process, configuration auditing and status reporting, SCM tools and best practices. Agile Software Management - agile values and iterative development principles, SCRUM framework: roles (Product Owner, Scrum Master, Development Team), ceremonies (sprint planning, daily standup, review, retrospective), and artifacts (product backlog, sprint backlog, burndown chart), Kanban methodology: Kanban boards, WIP limits, and flow management, Lean approaches: waste elimination, value stream mapping, and continuous improvement (Kaizen).

Text Books:

1. Sommerville, I. (2020). Engineering software products: An introduction to modern software engineering. Pearson Education.
2. Pressman, R. S. (2014). Software engineering: A practitioner's approach (8th ed.). McGraw Hill.
3. Anderson, D. J. (2003). Agile management for software engineering. Pearson.

Reference Books:

1. Rumbaugh, J., & Blaha, M. (2007). Object-oriented modeling and design with UML. Pearson Education.
2. Wang, Y. (2008). Software engineering foundations: A software science perspective. Auerbach Publications.
3. Denning, S. (2018). The age of agile. Amacom.

END SEMESTER EXAMINATION – MODEL QUESTION PAPER

Course Code	25UAMT411
Course Name	SOFTWARE ENGINEERING
Max. Marks: 60	Duration: 2.5 Hours

Part A (Answer all questions. Each question carries 3 marks)				
Qn No.	Question	Marks	CO	BL
1.	Compare the Waterfall and Spiral software process models with respect to their applicability, advantages, and limitations.	3	CO1	L2
2.	Explain the key values and principles of the Agile Manifesto. How does Extreme Programming implement these principles?	3	CO1	L2
3.	What is a Software Requirements Specification (SRS)? List and explain the characteristics of a good SRS document.	3	CO2	L2
4.	Explain the concept of cohesion and coupling. Why is high cohesion and low coupling desirable in software design?	3	CO2	L2
5.	Explain the MVC design pattern with a suitable example. How does it separate concerns in a web application?	3	CO3	L2

6.	Distinguish white-box and black-box testing. Explain the boundary value analysis technique with an example.	3	CO3	L2
7.	Explain the SCRUM framework in detail: roles, ceremonies, and artifacts. How does a sprint backlog differ from a product backlog?	3	CO4	L2
8.	What is CI/CD? Explain the stages of a typical CI/CD pipeline and the tools used at each stage.	3	CO4	L2

Part B (Answer any one full question from each Module. Each carry 9 marks)				
Module I				
Qn No.	Question	Marks	CO	BL
9.1	Explain the Incremental and Prototype software development models. Compare them with respect to flexibility, risk management, and client involvement.	5	CO1	L2
9.2	Describe the insulin pump control system case study. Illustrate the critical safety requirements and how are they captured in the software?	4	CO1	L3
OR				
10.1	Explain the software process activities: specification, design, validation, and evolution. How do these map to the Agile model?	5	CO1	L2
10.2	Describe different types of feasibility analysis. Write a feasibility report outline for developing an online examination system.	4	CO1	L3
Module II				
Qn No.	Question	Marks	CO	BL
11	For an online banking system, develop: (i) Use case diagram with at least 5 use cases, (ii) Class diagram with 4 entities and their relationships, (iii) Sequence diagram for a fund transfer operation. Explain each diagram.	9	CO2	L3
OR				
12	Explain the concepts of requirement elicitation techniques: interviews, questionnaires, and observation. Which is most effective for a safety-critical system?	9	CO2	L3
Module III				
Qn No.	Question	Marks	CO	BL
13	Explain the software testing strategy: unit, integration, validation, and system testing. Describe test-driven development (TDD) methodology and compare it with traditional testing.	9	CO3	L3

	Design test cases using boundary value analysis and equivalence partitioning for a function that accepts a number between 1 and 100.			
OR				
14	Apply DevOps practices to implement a CI/CD pipeline for a software development project. Demonstrate how Git version control supports collaboration, code integration, and deployment throughout the DevOps workflow.	9	CO3	L3
Module IV				
Qn No.	Question	Marks	CO	BL
15	A software development team is working on a collaborative project with frequent code updates. Apply the Software Configuration Management (SCM) activities—configuration identification, version control, change control, and status accounting—to manage the project effectively. Choose suitable SCM tools and explain how they support each activity.	4	CO4	L3
OR				
16	Describe the complete SCRUM framework in depth. A software team is developing a mobile app. Define: product backlog items, sprint goal, sprint ceremonies, and a sample burndown chart for a 2-week sprint. Also explain how Lean principles like Kaizen can be applied to improve the sprint retrospective.	9	CO4	L3