SWARNANDHRA

COLLEGE OF ENGINEEERING AND TECHNOLOGY (AUTONOMOUS)

SEETHARAMPURAM, NARSAPUR-534280, WG-DT, AP

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

TEACHINGPLAN

Course Code	Course Title	Year/ Sem.	Branch	Contact Hr./ week	Academic Year	Date of Commencement of Semester
24MC2T04	Software Engineering	I/II	MCA	6	2025-26	17.02.2025

COURSE OUTCOMES: Upon the successful completion of this course the student will be able

1. Describe the history and characteristics of Software Engineering and Compare SDLC methodologies like Waterfall, Spiral and Agile. (K2)
2. Apply techniques to gather and analyze software requirements and create a Software Requirement Specification(SRS).(K3)
3. Analyze design strategies and evaluate software design using metrics such as coupling and cohesion.(K4)
4. Develop and execute test cases using various techniques to ensure software quality.(K6)
5. Evaluate maintenance models and Re-engineering techniques to manage software Evaluation and upkeep.(K5)

UNIT	Outcome Blooms Level	TOPIC/ACTIVITY		Text Book	Contact Hours	Delivery Method
I		UNIT-I				Chalk &Board, PPT
		1.1	Software Engineering and history	T1	2	
		1.2	Software Crisis	T1	1	
		1.3	Evolving of a programming system product	T1	1	
		1.5	SDLC: Software Development Process	T1	2	

	Describe the history and characteristics of Software Engineering and Compare SDLC methodologies like Waterfall, Spiral and Agile.	1.6	Code-and-Fix model	T1	1	
		1.7	Waterfall model	T1	1	
		1.8	Evolutionary model	T1	1	
		1.9	Incremental Implementation, Prototyping	T1	1	
		1.10	Spiral model and Software reuse	T1	2	
		1.11	Critical Comparisons of SDLC models	T1	1	
		1.12	Introduction to Non-Traditional SDP: Rational Unified Process	T1	1	
		1.13	Rapid Application Development, Agile Development Process Introduction	T1	1	
		1.14	Agile – SCRUM Sprint, Review, Retrospective, Planning	T1	1	
		1.15	XP, KANBAN, SAFE Agile	T1	1	
II	Apply techniques to gather and analyze software requirements and create a Software Requirement Specification(SRS)	**UNIT–II**				Chalk Board & PPT, Video
		2.1	Requirements: Importance of Requirement Analysis, user needs	T1	1	
		2.2	Software Features and Software Requirements	T1	1	
		2.3	Classes of user Requirements: Enduring and volatile	T1	2	
		2.4	Sub-phases of Requirement Analysis	T1	1	
		2.6	Barriers to Eliciting user Requirements	T1	1	
		2.7	The software Requirements Document SRS Standards.	T1	1	
		2.8	Requirements Engineering, Case Study of SRS for a real time system	T1	2	

		2.9	Tools for Requirements Gathering: Document flow chart	T1	1	
		2.10	Decision Table, Decision Tree	T1	2	
		2.11	Introduction to Non – Traditional requirements	T1	1	
III	Analyze design strategies and evaluate software design using metrics such as coupling and cohesion	**UNIT–III**				Chalk &Board PPT Demonstration
		3.1	Software Desire: Goals of good software design	T1	1	
		3.2	Design Strategies and methodologies	T1	1	
		3.3	Data – Oriented software design	T1	1	
		3.4	Structured Design: Structure chart	T1	2	
		3.5	Coupling, Cohesion, Modular Structure	T1	1	
		3.6	Packaging, Object – Oriented Design	T1	1	
		3.7	Top – Down and Bottom – Up Approach, Design Patterns	T1	2	
		Mid I Exam				
		3.8	Structured Analysis: DFD, Data Dictionary	T1	1	
		3.9	Software Measurement and Metrics	T1	1	
		3.10	Various Size – Oriented Measures: Halstead's Software Science	T1	1	
		3.11	Function Point(FP) Based Measures, Cyclomatic Complexity Measures	T1	2	
		3.12	Control Flow Graphs Development	T1	1	
		3.13	Selecting a Language, Coding Guidelines	T1	1	

		UNIT-IV				
IV	Develop and execute test cases using various techniques to ensure software quality	4.1	Software Testing: Testing process, Design of Test cases	T1	1	Chalk & Board, PPT Demonstration
		4.2	Functional Testing:	T1	1	
		4.3	Boundary Value Analysis	T1	1	
		4.4	Equivalence Class Testing, Decision table testing	T1	1	
		4.5	Cause – Effect Graphing, Structural Testing	T1	1	
		4.6	Path Testing, Data Flow and Mutation testing	T1	2	
		4.7	Unit Testing	T1	1	
		4.8	Integration and System testing, Debugging	T1	1	
		4.9	Alpha & Beta Testing	T1	1	
		4.10	Testing Tools	T1	1	
		4.11	Standards	T1	1	
		UNIT-V				
V	Evaluate maintenance models and Re-engineering techniques to manage software Evaluation and upkeep	5.1	Software Maintenance	T1	1	Chalk &Board PPT
		5.2	Management of maintenance	T1	1	
		5.3	Maintenance process	T1	1	
		5.4	Maintenance models	T1	1	
		5.5	Regression Testing	T1	1	
		5.6	Reverse Engineering, Software Engineering	T1	1	
		5.7	Configuration Management	T1	1	
		5.8	Documentation	T1	1	
		MID EXAM 2				
		TOTALCLASSES			65	

Recommended Text Books for Reading:

1. Software Engineering: A Practitioner's Approach (9th Edition) by R. S. Pressman, McGraw Hill.

Reference Text Books:

1. Software Engineering by K. K. Aggarwal & Yogesh Singh, New Age International Publishers (3rd Edition, 2007)
2. Software Engineering by Ian Sommerville, Addison Wesley (9th Edition, 2010)
3. An Integrated Approach to Software Engineering by Pankaj Jalote, Narosa Publishing House (3rd Edition, 2007)

Faculty **Head of the Department** **Principal**