Course Title	**Course Structure**			**Pre-Requisite**
Software Engineering Methodologies	**L**	**T**	**P**	**Fundamentals of SE**
	3	**0**	**2**	

Course Objective:

To introduce the concepts of software engineering including requirement specifications, software design, testing and maintenance.

Course Outcome (CO):

1. Explain various software characteristics and analyse different software Development Models
2. Demonstrate the contents of a SRS and apply basic software quality assurance practices to ensure that design, development meet or exceed applicable standards
3. Compare and contrast various methods for software design
4. Formulate testing strategy for software systems, employ techniques such as unit testing, Test driven development and functional testing
5. Manage software development process independently as well as in teams and make use of Various software management tools for development, maintenance, and analysis

S.No.	Content	Contact Hours
Unit 1	Software Requirement Specification: Requirements Elicitation Techniques, Requirements analysis, Models for Requirements analysis, requirements specification, requirements validation.	9
Unit 2	System Design: Design Principles: Problem partitioning, abstraction. Top down and bottom up – design, structured approach. Functional versus 6object oriented approach of design, design specification, Cohesiveness and Coupling. Overview of SA/SD Methodology, structured analysis, data flow diagrams, extending DFD to structure chart.	8
Unit 3	Software project Management: Project planning and Project scheduling. Software Metrics: Size Metrics like LOC, Token Count, Function Count. Cost estimation using models like COCOMO. Risk management activities. Software Reliability and Quality Assurance: Reliability issues, Reliability metrics, reliability models, Software quality, ISO 9000 certification for software industry, SEI capability maturity model.	10
Unit 4	Testing: Verification and validation, code inspection, test plan, test case specification. Level of testing: Unit, Integration Testing, Top down and bottom-up integration testing, Alpha and Beta testing, System testing and debugging. functional testing, structural testing, Software testing strategies.	10

Unit 5	Software Maintenance: Structured Vs unstructured maintenance, Maintenance Models, Configuration Management, Reverse Engineering, Software Re-engineering. Case study: Practical applications of SDLC phases	5
	Total	42

Books:-

S.No.	**Name of Books/Authors/Publisher**
1	R. S. Pressman, “Software Engineering – A practitioner’s approach”, 3rd ed., McGraw Hill Int. Ed..
2	K. K. Aggarwal & Yogesh Singh, “Software Engineering”, 2ndEd., New Age International.
3	Sommerville, “Software Engineering”, Addison Wesley.