Software Engineering

Course Code	20CS3501	**Year**	III	**Semester**	I
Course Category	PCC	**Branch**	CSE	**Course Type**	Theory
Credits	3	**L-T-P**	3-0-0	**Prerequisites**	OOP
Continuous Evaluation :	30	**Semester End Evaluation:**	70	**Total Marks:**	100

Course Outcomes		
Upon successful completion of the course, the student will be able to		
CO1	Understand the fundamentals of software engineering	**L2**
CO2	Apply various lifecycle activities for project development	**L3**
CO3	Apply Risk and Quality Management Strategies for project development	**L3**
CO4	Analyze the various requirements, design and Testing Techniques to select the appropriate techniques for the software project development	**L4**

Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3:Substantial, 2: Moderate, 1:Slight)

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3													
CO2	3									1				
CO3										1			2	
CO4		3							1	1				

Syllabus		Mapped CO
Unit No.	**Contents**	
I	Introduction to Software Engineering: The Nature of Software, The changing Nature of Software, Software Process, Software Engineering Practice, Software Development myths. Process models: Perspective process Models, Unified Process Model, **Agile process model**	**CO1,CO4**
II	Requirements engineering: Requirements engineering, Establishing the Groundwork, Eliciting Requirements, Developing Use Cases, Building the Analysis Model, Negotiating requirements. Agile Requirements: User Stories, Backlog Management. Requirements Analysis: Scenario Based Modeling.	**CO1,CO2,CO4**
III	Design Engineering: Design process, Design concepts, The Design Model. Architectural design: Software architecture, Architectural styles, Architectural Design, Agility and Architecture User interface design: Golden rules	**CO1,CO2,CO4**
IV	Software Testing Strategies: A strategic approach to software testing, Test strategies for conventional software, Test strategies for Object-Oriented software Validation Testing, System Testing, The art of Debugging. Testing Conventional Applications: Software testing fundamentals, White-Box testing, Basis path testing, Control structure testing, Black-Box testing Agile Testing, User Acceptance Test.	**CO1,CO2,CO4**
V	Risk management: Reactive vs. Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM Plan. Software Quality Assurance: Elements of SQA, SQA Tasks, Goals, and Metrics, Statistical Software Quality Assurance, Software Reliability, The ISO 9000 Quality Standards, The SQA Plan. Agile approach to Risk and Quality Assurance	**CO1,CO3,CO4**

Learning Resources
Text Books
1. Software Engineering: A Practitioner's Approach, Roger S. Pressman, Bruce R. Maxim, Eighth edition, 2015, McGraw Hill, International Edition 2. Robert C. Martin ,Agile Software Development, Principles, Patterns, and Practices Alan Apt Series (2011)
References
1. Software Engineering, Ian Sommerville, Seventh edition, 2004, Pearson, India 2. Software Engineering, K.K. Agarwal &Yogesh Singh, 2007, New Age International Publishers. 3. Software Engineering Principles and Practice, Waman S Jawadekar, 2004, McGraw-Hill. 4. Fundamentals of Software Engineering, Rajib Mall, Fourth edition, 2009, PHI. 5. Succeeding with Agile : Software Development Using Scrum, Pearson (2010)
e-Resources & other digital material
1.https://onlinecourses.nptel.ac.in/noc20_cs68 *2.https://thedigitalprojectmanager.com*