Course Code	Course Title							
116U01C501	Software Engineering							
	TH			P		TUT		Total
Teaching Scheme(Hrs.)	03			--		--		03
Credits Assigned	03			--		--		03
Examination Scheme	Marks							
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	30	20	50	--	--	--	--	100

Course prerequisites (if any):

-

Course Objectives:

The Course focusses at developing an understanding of software process models such as the waterfall and evolutionary models. It Further provides, an understanding of software requirements and the SRS documents. The course aims at enabling the students to prepare the system design and test cases for proper testing of the software.

Course Outcomes:

At the end of successful completion of the course the student will be able to

CO1 : Understand the software development process and Estimate different types of resources for the given project.
CO2 Analyze the software requirements and Model the defined problem with the help of UML diagram.
CO3 Prepare the System Design and Model
CO4 Identify and manage configuration items and risks for the software
CO5 Test the given software for different test cases with proper test planning.

Module No.	Unit No.	Details	Hrs.	CO
1	**The Product and the Process:**		08	CO 1
	1.1	Software life cycle models: Waterfall, RAD, Spiral, Agile process.		
	1.2	Understanding software process, Process metric, CMM Levels		
	1.3	Planning & Estimation: Product metrics Estimation-LOC, FP, COCOMO models.		
	1.4	Project Management activities : Planning, Scheduling and Tracking		
2	**Requirement Engineering**		08	CO 2
	2.1	Introduction to OO Methodologies :Booch,Rambaug and Jacobson		
	2.2	Requirements Engineering Tasks, Requirement Elicitation Techniques, Software Requirements: Functional, Non- Functional		
	2.3	Requirements Characteristics, Requirement qualities, Requirement Specification, Requirement Traceability, System Analysis Model Generation, Documentation : Use Case Diagram, Acitvity Diagram		
	2.4	Categorizing classes: entity, boundary and control ,Modeling associations and collections-Class Diagram		
	2.5	Dynamic Analysis - Identifying Interaction – Sequence and Collaboration diagrams, State chart diagram		
3	**System Design Engineering**		7	CO 3
	3.1	Design quality, Classification of Design Activities, Design Concepts: Modularity and Layering, Introduction to Pattern-Based Software Design,		
	3.2	Software Architecture, Data Design, Object-Oriented versus Function-Oriented Design, Design of Software Objects, Methods, Cohesion and Coupling between Objects,		
	3.3	User Interface Design: Rules, User Interface Analysis and Steps in Interface Design, Design Evaluation		
	3.5	Software Reuse, Component-Based Software Engineering		
4	**System Implementation, Configuration Management & Risk Management**		14	CO 4

	4.1	Packages and interfaces: Distinguishing between classes/interfaces, Exposing class and package interfaces		
	4.2	Mapping model to code , Mapping Object Model to Database Schema		
	4.3	Component and deployment diagrams: Describing Dependencies		
	4.4	Managing and controlling Changes, Managing and controlling version		
	4.5	Categories of Risks, Nature Of Risk, Types of Risk, Risk Identification, Risk Assessment, Risk planning and control, Risk management, Evaluating risk to schedule, PERT technique.		
5	**Testing and Maintenance**		**8**	**CO 5**
	5.1	Testing Concepts: Purpose of Software Testing, Testing Principles, Goals of Testing, Testing aspects: Requirements, Test Scenarios, Test cases, Test scripts/procedures,		
	5.2	Strategies for Software Testing, Testing Activities: Planning Verification and Validation, Software Inspections,FTR		
	5.3	Levels of Testing : unit testing, integration testing, regression testing, product testing, acceptance testing and White-Box Testing		
	5.4	Black-Box Testing: Test Case Design Criteria, Requirement Based Testing, Boundary Value Analysis, Equivalence Partitioning		
	5.5	Object Oriented Testing: Review of OOA and OOD models, class testing, integration testing, validation testing		
	5.6	Reverse and re-engineering, types of maintenance		
	#Self-Learning : Testing tools			
Total			**45**	

#Students should prepare all Self Learning topics on their own. Self-learning topics will enable students to gain extended knowledge of the topic. Assessment of these topics may be included in IA and Laboratory Experiments.

Recommended Books:

Sr. No.	Name/s of Author/s	Title of Book	Name of Publisher with country	Edition and Year of Publication
1	Roger Pressman	Software Engineering	Tata McGraw Hill.	Sixth edition, 2010
2	Bernd Bruegge	Object oriented software engineering	Pearson Education.	Third Edition, 2009
3	Ian Sommerville	Software Engineering	Pearson Education	Sixth edition, 2001
4	John Nicholas, Herman Steyn	Project Management for Business Engineering and Technology	Routledge	5th Edition, 2017
5	Bob Hughes, Mike cotterell, Rajib Mall	Software Project Management	Tata McGraw Hill	fifth Edition, 2012