

Course Code	21CSC303J	Course Name	SOFTWARE ENGINEERING AND PROJECT MANAGEMENT	Course Category	C	PROFESSIONAL CORE	L	T	P	C
							2	0	2	3

Pre-requisite Courses	Nil	Co-requisite Courses	Nil	Progressive Courses	Nil
Course Offering Department	School of Computing	Data Book / Codes / Standards			Nil

Course Learning Rationale (CLR):		The purpose of learning this course is to:		Program Outcomes (PO)												Program Specific Outcomes		
CLR-1:				1	2	3	4	5	6	7	8	9	10	11	12			
CLR-1:	familiarize the software life cycle models and software development process			Engineering Knowledge	Problem Analysis	Design/development of solutions	Conduct investigations of complex problems	Modern Tool Usage	The engineer and society	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO-1	PSO-2	PSO-3
CLR-2:	illustrate the various techniques for requirements, planning and managing a technology project																	
CLR-3:	examine basic methodologies for software design, development, testing, and implementation																	
CLR-4:	understand manage user's expectations and the software development team																	
CLR-5:	apply the project management and analysis principles to software project development																	
Course Outcomes (CO):		At the end of this course, learners will be able to:																
CO-1:	identify the process of project life cycle model and process			-	-	-	-	-	-	-	-	2	-	2	-	3	-	-
CO-2:	analyze and translate end-user requirements into system and software requirements			-	3	-	-	-	-	-	-	2	-	2	-	3	-	-
CO-3:	identify and apply appropriate software architectures and patterns to carry out high level design of a system			-	-	2	-	-	-	-	-	2	-	2	-	3	-	-
CO-4:	develop Test plans and incorporate suitable testing strategies			-	-	-	-	-	-	-	-	2	-	2	-	3	-	-
CO-5:	examine the risk strategies and maintenance measures			-	-	-	-	-	-	-	-	2	-	3	-	3	-	-

<b>Unit-1 - Introduction to Software Engineering</b>	<b>12 Hour</b>
The evolving role of software, changing nature of software, Generic view of process: Software engineering- a layered technology, a process framework, Software Project Management - life cycle activities, Process models: The waterfall model, incremental process models, evolutionary process models, the unified process, Conventional- Agile, XP, Scrum, Project Initiation management – Project Charter, Project Scope, Project Objectives, Practical considerations.	
<b>Unit-2 - Software Requirements</b>	<b>12 Hour</b>
Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document. Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management, Software project effort and cost estimation – Cocomo model I, Cocomo Model II, LOC, Function point metrics	
<b>Unit-3 - Software Design</b>	<b>12 Hour</b>
Software Design Fundamentals, Design process – Design Concepts-Design Model– Design Heuristic , Design techniques– Architectural Design - Architectural styles, Creating an architectural design- software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams, Design of User Interface design Elements of good design, Design issues Features of modern GUI - Menus, Scroll bars, windows, Buttons, icons, panels, error Messages etc.	
<b>Unit-4 - Software Construction</b>	<b>12 Hour</b>
Coding Standards, Coding Frameworks. Reviews: Deskchecks, Walkthroughs, Code Reviews, Inspections, Coding Methods, Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, Unit Testing – Integration Testing – Validation Testing – System Testing and Debugging	
<b>Unit-5 – Product Management</b>	<b>12 Hour</b>
Product Release Management, Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan, Maintenance and Reengineering	

Lab Experiments	
<p>Lab 1: Identify the Software Project, Create Business Case, Arrive at a Problem Statement</p> <p>Lab 2: Analyse Stakeholder and User Description and Identify the appropriate Process Model</p> <p>Lab 3: Identify the Requirements, System Requirements, Functional Requirements, Non-Functional Requirements and develop a SRS Document</p> <p>Lab 4: Prepare Project Plan based on scope, Find Job roles and responsibilities, Calculate Project effort based on resources</p> <p>Lab 5: Prepare the Work, Breakdown Structure based on timelines, Risk Identification and Plan</p> <p>Lab 6: Design a System Architecture, Use Case Diagram, ER Diagram (Database)</p> <p>Lab 7: DFD Diagram (process) (Upto Level 1), Class Diagram (Applied For OOPS based Project),</p> <p>Lab 8: Interaction Diagrams, State chart and Activity Diagrams</p> <p>Lab 9: State and Sequence Diagram, Deployment Diagram,</p> <p>Lab 10: Sample Frontend Design (UI/UX)</p> <p>Lab 11: Sample code implementation</p> <p>Lab 12: Master Test Plan, Test Case Design (Phase 1</p> <p>Lab 13: Manual Testing</p> <p>Lab 14: User Manual, Analysis of Costing, Effort and Resource</p> <p>Lab 15: Project Demo and Report Submission with the team</p>	

Learning Resources	Learning Resources	
	1. Roger S. Pressman, Software Engineering – A Practitioner Approach, 6th ed., McGraw Hill, 2005	4. Ramesh, Gopalswamy, Managing Global Projects, Tata McGraw Hill, 2005
	2. Ian Sommerville, Software Engineering, 8th ed., Pearson Education, 2010	5. Ashfaq Ahmed, Software Project Management: a process-driven approach, Boca Raton, Fla: CRC Press, 2012
	3. Rajib Mall, Fundamentals of Software Engineering, 4th ed., PHI Learning Private Limited, 2014	6. Walker Royce, Software Project Management, Pearson Education, 1999
		7. Jim Smith Agile Project Management: Creating Innovative Products, Pearson 2008

Learning Assessment							
	Bloom's Level of Thinking	Continuous Learning Assessment (CLA)				Summative Final Examination (40% weightage)	
		Formative CLA-1 Average of unit test (45%)		Life-Long Learning CLA-2 (15%)			
		Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember	20%	-	-	20%	20%	-
Level 2	Understand	20%	-	-	20%	20%	-
Level 3	Apply	40%	-	-	40%	40%	-
Level 4	Analyze	20%	-	-	20%	20%	-
Level 5	Evaluate	-	-	-	-	-	-
Level 6	Create	-	-	-	-	-	-
	Total	100 %		100 %		100 %	

Course Designers		
Experts from Industry	Experts from Higher Technical Institutions	Internal Experts
1. Mr. DHINAKAR JACOB SELWYN, CAP GEMINI TECHNOLOGY		1. Mrs. Anupama C G, SRMIST
2. Mr. Girish Raghavan, Wipro Technologies		