

Branch Name:	Computer Engineering / Information Technology
Program Code:	ET103 / ET105
Course Title:	Software Engineering
Course Code:	2ET1030501T
Pre-requisite Course:	Object Oriented Programming fundamental, UML. Though, awareness about software systems, computer fundamentals would be beneficial.

Course Objective:

- Be successful professionals in the field with solid fundamental knowledge of software engineering.
- Utilize and exhibit strong communication and interpersonal skills, as well as professional and ethical principles when functioning as members and leaders of multi-disciplinary teams.
- Apply their foundations in software engineering to adapt to readily changing environments using the appropriate theory, principles, and processes.

Teaching and Examination Scheme:

Credit	Teaching Scheme Hours per week				Evaluation Scheme (Marks)								Total (Marks)
					Theory (Marks)				Practical (Marks)				
	L	T	P	Total	SEE		CCA		SEE		CCA		
					Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
3	3	-	-	3	24	60	16	40	-	-	-	-	100

Course Contents:

Sr. No	Topic	Total Hours	Weightage (%)
1	Introduction to Software Engineering: The evolving role of software, Changing Nature of Software, legacy software, Software myths. A Generic view of process: Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), Process patterns, process assessment, personal and team process models.	06	15
2.	Process Models in Software Engineering: A Layered Technology, Software Process Models, The Linear Sequential Model, The Prototyping Model, The RAD Model, Agile Process Model, Component-Based Development, The water fall model (classical life cycle model), Incremental Process model, Evolutionary process model.	07	20
3.	Design and Modeling with UML: Software design concepts, Coupling and cohesion, UML diagrams: Use case diagram, class diagram, data flow diagram, activity diagram, Data dictionary, Software designs: Function oriented design and object oriented design, Software design approaches: top down and bottom up approach Architectural design, Component-Level design, User-Interface Design, Pattern Based Design, Web-App Design.	07	20
4.	Software Testing: Software testing concept, designing of test cases and Testing techniques, Unit testing, black box testing, white box testing, testing web and mobile application, Testing tools (win runner, load runner), testing documentation.	06	15
5.	Risk management: Reactive vs Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM Plan. Quality Management: Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards	06	15

6.	Advanced Topics in Software Engineering: Component-Based Software Engineering, Client/Server Software Engineering, Web Engineering, engineering, Computer-Aided Software Engineering, Software Process Improvement, and Emerging Trends in software Engineering.	05	15
----	---	----	----

Reference books:

- Roger S. Pressman, Software engineering- A practitioner's Approach, McGraw-Hill International
- Editions Software Engineering by Sommerville
- Fundamentals of Software Engineering by Rajib Mall
- Fundamentals of Software Engineering by K. K. Agrawal
- Object-Oriented Modelling and Design with UML by Rumbaugh, Blaha
- Ian Sommerville, Software engineering, Pearson education Asia
- Software Engineering Fundamentals by Ali Behhforoz & Frederick Hudson OXFORD
- John M Nicolas, Project Management for Business, Engineering and Technology, Elsevier

e-Resources:

- <http://nptel.ac.in/>
- <http://www.mhhe.com/engcs/compsci/pressman/student/olc/cases.mhtml>
- <http://www.softwaretestingclass.com/software-testing-tools-list/>
- <http://www.rspa.com/spi/>
- <https://www.edx.org/school/uc-berkeleyx>

Course Learning Outcomes (CLO):

Course Outcomes	Description	Bloom's Taxonomy Level
COL1	Design the application of different SDLC models for different types of projects.	Understanding (2) Applying (3)
COL2	Develop and understand the application of different requirement models require for the any software.	Understanding (2) Applying (3)
COL3	Gather, understand, analyses and document the software requirements.	Remember (1) Understanding (2)
COL4	Manage the software project development by staffing, scheduling and estimation techniques.	Remember (1) Understanding (2)
COL5	Understand the fundamental of risk management and Quality management	Remember (1) Understanding (2)
COL6	Understand the advance and emerging trends of software engineering.	Understanding (2)

Mapping of CLO with PO and PSO

Course Learning Outcomes	Program Outcomes (POs)												Program Specific Outcomes(PSOs)			
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3	PSO 4
CLO1	H	M	H		M	M		H	M	M	M	H				
CLO2	H	M	M		M	M		H	M	L	L	L				
CLO3	H	M	M		L	L		M	M	H	L	H				
CLO4	H	M	M		M	L		H	M	H	L	L				
CLO5	H	M	L		M	M		M	M	L	L	L				
CLO6	M	M	L		M	M		M	M	L	L	L				

H: High, M: Medium, L: Low