CS6304	SOFTWARE ENGINEERING	L	T	P	EL	TOTAL CREDITS
		3	0	0	3	4

Prerequisites for the course: None

OBJECTIVES:
- To gain knowledge about various software development lifecycle (SDLC) models
- To appreciate the importance of requirements engineering in SDLC
- To be aware of designing a software, considering the various perspectives of the end user
- To learn to develop a software component using coding standards and facilitate code reuse
- To analyze the software using metrics and measurements and predict the complexity and the risk associated
- To appreciate appropriate software documentations across various SDLC stages

MODULE I:	L	T	P	EL
	4	0	0	3

Introduction –Product and Process –Software Development Lifecycle –Waterfall Model – Incremental Models – Evolutionary Models - Spiral Model – Unified model. Software Project Planning

SUGGESTED ACTIVITIES:
- In-class activity on Application specific Product and Process view
- External Learning on impact of unified process models on Quality Software Development and JIT software

SUGGESTED EVALUATION METHODS:
- Assignments: Selection of suitable software process models for a given software specification
- Tutorial problems: Identification of Sample Application for each process model and justify the same stating reasons.

MODULE II:	L	T	P	EL
	3	0	0	3

CMM – CMMI – PSP – TSP – ISO 12207 (Software Lifecycle), ISO 29148 (Requirements), ISO 15026 (Risk & Integrity), ISO 29119 (Testing), ISO 14764 (Maintenance), ISO 15939 (Measurement)

SUGGESTED ACTIVITIES:
- Need for organization wide standards adoption

SUGGESTED EVALUATION METHODS:
- Recalling the KPAs to be adhered for each level in CMM.
- Assignment on selection of appropriate standards for each phase in software development.

MODULE III:	L	T	P	EL
	3	0	0	3

Requirements Elicitation – Analysis & Negotiation – Requirements Modeling & Specification – Requirements Validation & Management

SUGGESTED ACTIVITIES:
- External Learning: Using open-source tools for RE to understand the requirements traceability and interdependency among the functionalities provided by the software project.

SUGGESTED EVALUATION METHODS:
- Tutorial on various Requirements elicitation mechanisms and selection of an appropriate strategy.

	L	T	P	EL

- Assignment on Requirements categorization (considering contradicting, omission, commission of requirements) in a software project

MODULE IV:	L	T	P	EL
	2	0	0	3

Data Modeling– Scenario Based Modeling

SUGGESTED ACTIVITIES:
- External Learning: Using open-source tools for Conceptual Data modeling of a Sample application
- External Learning: Using open-source tools for Scenario based modelling of a problem statement.

SUGGESTED EVALUATION METHODS:
- Assignment Data Modeling of sample application
- Assignment: Designing use case diagram and activity diagram to analyze the requirements obtained from the customer and segregate them as use cases and determine the possible set of activities from the end user.

MODULE V:	L	T	P	EL
	3	0	0	3

Flow Oriented Modeling – Class Based Modeling – User Interface Design

SUGGESTED ACTIVITIES:
- External Learning: Use open source tools to perform Class Based Modeling for a given software requirements.

SUGGESTED EVALUATION METHODS:
- Assignment: Determine the flow of data/events among the processes in the application under consideration
- Assignment: Designing UI of Sample application
- Assignment: Design-to-code of Sample application involving coding standards

MODULE VI:	L	T	P	EL
	3	0	0	3

Testing strategies – Unit testing, Integration testing, OO testing – Smoke testing – Validation testing – System testing - Alpha/Beta testing – Recovery Testing – Security Testing – Stress Testing – Performance Testing

SUGGESTED ACTIVITIES:
- External Learning: Understanding the requirements (SRS) and designing a suitable test suite.
- External Learning: Determine valid interfaces for integration testing and design necessary stub and driver modules
- External Learning on ideas of testing a simple online application on selected test cases
- Tutorial on using Automation software for testing

SUGGESTED EVALUATION METHODS:
- Assignment on obtaining a mind-map on testing strategies
- Assignment: Testing of Sample application using any OSS on Software Test Automation

MODULE VII:	L	T	P	EL
	3	0	0	3

Debugging Process - Testing Tactics – Black-box approaches – Graph based testing Methods – Equivalence class partitioning – Boundary value analysis – Orthogonal array testing

SUGGESTED ACTIVITIES:
- In-class activity on Equivalence class partitioning
- In-class activity on Boundary value analysis

	• External Learning on Software Test Documentation				

SUGGESTED EVALUATION METHODS:
- Assignment: Testing Sample application using Black-box approaches and understand the differences in selecting of test cases from the test suite.

MODULE VIII:	**L**	**T**	**P**	**EL**
	3	**0**	**0**	**3**

Testing Tactics – White-box approaches– Basis Path testing – Control Structure Testing

SUGGESTED ACTIVITIES:
- In-class activity on Basis Path testing
- In-class activity on Control-structure testing

SUGGESTED EVALUATION METHODS:
- Assignment: Testing Sample application for White-box approaches and understand how it differ from black box testing approaches.

MODULE IX:	**L**	**T**	**P**	**EL**
	3	**0**	**0**	**3**

Software Maintenance –Software Project Management - Software Cost Estimation - Risk Management& Mitigation – Configuration Management – Software Documentation standards

SUGGESTED ACTIVITIES:
- External Learning on using tools for estimating Software Cost

SUGGESTED EVALUATION METHODS:
- Tutorial: Identification of potential risks for a software project during development/ maintenance and tabulate.
- Assignment: Using a Software Configuration Management template for a software project

MODULE X:	**L**	**T**	**P**	**EL**
	3	**0**	**0**	**3**

Software Metrics – Process, Project and Product Metrics – OO Metrics - Test Metrics: Test effort, effectiveness, efficiency metrics – Test coverage, Test execution & Defect Distribution Metrics

SUGGESTED ACTIVITIES:
- External Learning on Software Quality Models
- In-class activity on FP metrics & Variants
- External Learning on Software Test Lifecycle

SUGGESTED EVALUATION METHODS:
- Assignment: Calculation of test metrics for sample application

OUTCOMES:
Upon completion of the course, the students will be able to:
- Point out the role and impact of software engineering in contemporary business, and global, economic, environmental and societal context
- Analyze and resolve information technology problems through the application of systematic approaches and diagnostic tools

- Analyze, design and manage the development of a computing-based system, component or process to meet the desired needs within realistic constraints in one or more application domains
- Use knowledge, techniques, skills and modern tools necessary for software engineering practice
- Engineer tools to analyze, evaluate, select and synthesize information sources for the purpose of developing a software system

TEXTBOOKS:
1. Roger S. Pressman, "Software Engineering: A Practitioner's Approach", McGraw Hill, Seventh Edition, 2009.
2. Ian Sommerville, "Software Engineering", Ninth Edition, Pearson Education, 2008.

REFERENCES:
1. Jalote Pankaj, "An Integrated Approach to Software Engineering", Third Edition, Springer, 2010.
2. Shari Lawrence Pfleeger and Joanne M. Atlee, "Software Engineering: Theory and Practice", Fourth Edition, Prentice Hall, 2010.

EVALUATION PATTERN:

Category of Course	Continuous Assessment	Mid – Semester Assessment	End Semester
Theory	40	20	40

CO - PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	
CO2	✓	✓			✓							
CO3	✓	✓			✓		✓		✓	✓		✓
CO4	✓	✓									✓	
CO5	✓	✓	✓		✓							