COURSE CONTENT

SOFTWARE ENGINEERING								
III Semester: CSD / CSE / CSM								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
		L	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
2430505	**Core**	2	0	0	2	40	60	100
Contact Classes: 30	**Tutorial Classes: Nil**	**Practical Classes: Nil**				**Total Classes: 30**		
Prerequisites: **There are no prerequisites to take this course.**								

Course Overview:

This course introduces the characteristics of software Generic view of software and the process models. The course emphasizes on software requirements system models, UML Basics, testing strategies, Metrics for process & products, instruction set design, RISK Management, RISK Identification, Quality Management, ISO : 9000 Quality.

Course Objectives: **The students will try to learn**

1. Importance of software engineering principles and software process framework
2. Contemporary approaches for design models and requirements validation
3. Various metrics and quality assurance strategies
4. Designing of testing report.
5. Different strategies for testing and risk management

Course Outcomes: After Completion of the Course, Students should be able to

- Explain the fundamental principles of software engineering, including software process models, agile methodologies, CMMI, and the evolving nature of software systems.
- Analyze software requirements using appropriate elicitation, validation, and management techniques, and prepare a structured Software Requirements Specification (SRS) document.
- Develop system models and design solutions using appropriate design principles, architectural styles, and UML diagrams for representing software structure and behavior.
- Apply software testing strategies and quality metrics for evaluating software products, ensuring verification, validation, and continuous process improvement.
- Assess software risks and implement quality management practices, including risk mitigation planning, software reviews, reliability evaluation, and adherence to ISO quality standards.

Module – I

INTRODUCTION TO SOFTWARE ENGINEERING: The Evolving Role of Software, Characteristics of Software, The 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.

Process models: The waterfall model, incremental process models, evolutionary process models, the unified process, Agile models: Extreme Programming, Scrum, DSDM, FDD, CRYSTAL and Lean Software Development

Module – II

Software Requirements: 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.

Module – III

System models: Context models, behavioral models, data models, object models, structured methods Design Engineering: Design process and design quality, design concepts, the design model, software architecture, Architectural styles and patterns. Introduction to UML: Basic Building Blocks of UML- Things, Relationships and Diagrams.

Module – IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, verification and validation testing, system testing, the art of debugging.

Metrics for Process and Products: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance, metrics for software quality.

Module – V

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, software reliability, the ISO 9000 quality standards.

TEXT BOOKS:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, McGraw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson Education.
3. The unified modelling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.

REFERENCE BOOKS:

1. Software Engineering, an Engineering approach- James F. Peters, WitoldPedrycz, John Wiley.
2. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies.
3. Fundamentals of object-oriented design using UML Meiler page-Jones: Pe

ELECTRONIC RESOURCES:

1. https://www.geeksforgeeks.org/software-engineering/software-engineering/
2. https://programiz.pro/resources/roadmap-to-become-a-software-engineer/
3. https://www.coursera.org/learn/introduction-to-software-engineering
4. https://www.tutorialspoint.com/software_engineering/index.htm
5. https://www.opit.com/magazine/software-engineering-2/

MATERIALS ONLINE:

1. Course template
2. Assignments
3. Model question paper – I
4. Model question paper – II
5. Lecture notes
6. E-Learning Readiness Videos