SUBJECT NAME: Software Engineering **Credit: 4 (3L + 1T)**
SUBJECT CODE: BCAC403

COURSE OBJECTIVE:

The course on Software Engineering aims to provide students with a comprehensive understanding of the principles, methodologies, and best practices essential for developing high-quality software systems. Throughout the course, students will delve into various aspects of the software development lifecycle, including requirements analysis, design, implementation, testing, deployment, and maintenance. By the end of the course, students will be proficient in designing, developing, and managing software systems that meet stakeholder requirements and industry standards.

COURSE OUTCOME	
CO1	Understand the fundamental principles and concepts of software engineering, including software development life cycle models, requirements engineering, and software design paradigms.
CO2	Analyze and apply various software development methodologies, such as Waterfall, Agile, Scrum, and DevOps, to plan, execute, and manage software projects effectively.
CO3	Apply software design principles, such as modularity, abstraction, encapsulation, and cohesion, to create maintainable, scalable, and extensible software solutions.
CO4	Perform software testing activities, including unit testing, integration testing, system testing, and acceptance testing, to ensure the quality, reliability, and correctness of software products.
CO5	Collaborate effectively in multidisciplinary teams to analyze, design, implement, and test software solutions for real-world problems, demonstrating effective communication, teamwork, and problem-solving skills

DETAILED SYLLABUS:

Module No:	NAME OF THE TOPIC	HOURS	L	T	MARKS
M1	Introduction: A Generic View of Software Engineering, Phases in software development, Linear Sequential Model, Prototype model, Evolutionary Model (Incremental and spiral model), Specialized Process Model (Component-Based Development, Aspect-Oriented Software Development), Agile Process (Principles, Human factors), Simple Case Study	4	3	1	6

M2	Project Metrics: Software Measurement (Size oriented, Function Oriented, Extended Function Point Metrics, Object-Oriented Metrics, Web application project Metric), metric for Software quality (Measuring Quality, Defect Removal efficiency), Integrate metric with software(Establishing a baseline)	4	3	1	5
M3	Object Oriented Concepts: class, Objects, attributes, Operations, Methods, and Services, Messages, Identifying the Elements of an Object Model (identification of class and objects, Defining Operations), Object Oriented analysis and Design (use cases, Class-Responsibility-Collaborator Modeling,), Object-Relationship Model	4	3	1	6
M4	Project planning : Identification of Software scope (Feasibility), Resource Identification (Human resource, Reusable Software Resources), Empirical Estimation Models (COCOMO model), Estimation for Object oriented project, Estimation for agile development, Estimation for Web application project, Estimation of human resource requirement, Team structure, Time estimation, Project scheduling (Time-Line Charts, Tracking the Schedule, Scheduling for WebApp Projects) Make / buy decision (Creating a Decision Tree), Project Monitoring Plan (Time sheet, reviews, Cost schedule Milestone graph), Risk management (Identification, Prioritization, Risk Mitigation, Monitoring, and Management)	15	12	3	15
M5	Software Architecture (Fan out and fan in structure), Structural Partitioning (Horizontal and vertical Partition), Functional Independence (Coupling and Cohesion), Design heuristics for effective modularity	2	2	0	4
M6	Design: data Design , Architectural design / mapping using data Flow(Transform flow and Transactional flow), Designing class Based Components, Component Level design for Web Application, User Interface design Technique and documentation (Reduce the User’s Memory Load, make interface consistent), Interface Design steps (User Interface Design Patterns, design issues), Discuss with a case study,	8	6	2	6

M7	Software Testing : Objective, Pimples, Test case Design for conventional software (Unit testing, Integration Testing), path testing , cyclometic complexity, Test Strategy for Object Oriented software, Test cases for web application, validation testing, System testing (Recovery testing, Security testing, stress testing, Performance testing)	10	7	3	14
M8	Quality Management: Define quality of software, McCall’s Quality Factors, ISO 9126 Quality Factors, Achieve software quality (Software Engineering Methods, project Management Technique), Quality Control, Quality Assurance (elements of quality assurance), SQA goals, tasks, Metrices, Six Sigma for Software Engineering	6	5	1	6
M9	Software Configuration Management (SCM): Elements of a Configuration Management System, Baseline, Software Configuration Items, SCM Features, SM processes (Version Control, Change Control, Configuration Audit, Status Reporting), SCM for Web application (WebApp Configuration Objects, Content management, Change management)	4	3	1	8
	INTERNAL EXAMINATION	3			30
	TOTAL	60	44	13	100

SUGGESTED READING:

1. "Software Engineering: Principles and Practices" by Deepak Jain, S. K. Gupta - Publisher: Laxmi Publications
2. "Software Engineering and Quality Assurance" by Kshirasagar Naik, Priyadarshi Tripathy - Publisher: Oxford University Press
3. "Software Engineering: Theory and Practice" by Shariq Mahmood, A. A. Sastry - Publisher: Oxford University Press
4. "Software Engineering: A Precise Approach" by Pankaj Jalote - Publisher: Wiley India
5. "Software Engineering" by Pankaj Jalote - Publisher: Pearson Education India
6. "Software Engineering: A Lifecycle Approach" by Surajit Ghosh, Anirban Basu - Publisher: Pearson Education India
7. "Fundamentals of Software Engineering" by Rajib Mall - Publisher: Prentice Hall India
8. "Software Engineering: A Practitioner's Approach" by Roger S. Pressman - Publisher: McGraw-Hill Education
9. "Software Engineering" by Ian Sommerville - Publisher: Pearson Education Limited
10. "Introduction to the Team Software Process" by Watts S. Humphrey - Publisher: Addison-Wesley Professional