

Subject Code	Subject Name	Credits
26CS114	SOFTWARE ENGINEERING	4

Course Objectives:

The objective of this course is to provide a strong foundation in software architecture and design patterns for building scalable, high-performance systems. Students will learn modern architectural styles, microservices, and cloud-native development alongside advanced methodologies. The course also covers critical quality attributes, DevOps practices, and CI/CD for effective software lifecycle management.

Learning Objectives:

After completion of the course the students will be able to:

1. Explain the basic concepts and importance of software engineering.
2. Identify and apply appropriate software development life cycle (SDLC) models.
3. Analyze and document software requirements using SRS.
4. Design software systems using UML diagrams and design principles.
5. Understand and manage software maintenance activities.

Unit 1 - Introduction to Software Engineering (12 hrs.)

What is Software Engineering - Software characteristics, Software Crisis, Software Development Life Cycle (SDLC) , Types of Software , Software Process Models: Waterfall Model , Spiral Model , Iterative Model ,Agile Model.

Unit 2 - Requirements Engineering (12 hrs.)

Requirement gathering, Requirement analysis Functional vs Non-functional requirements, Software Requirement Specification (SRS), Feasibility study, Requirement validation.

Unit 3 - System Design (12 hrs.)

Design concepts -Importance of modular design in software engineering, Coupling and cohesion, Architectural design - Client-server architecture in modern applications, Layered architecture in software systems, Data design - Interface design, UML diagrams - Use case diagram, Class diagram, Sequence diagram.

Unit 4- Software Testing (12 hrs.)

Software Testing- principles of software testing, Software Quality and Testing relationship, Testing basics - Types of testing- Different levels of software testing explained, Functional testing vs Non-functional testing, Importance of testing in SDLC, Unit testing, Integration testing , System testing , Black-box testing , White-box testing , Debugging.

Unit 5 -Maintenance & Project Management (12 hrs.)

Software maintenance, Types of maintenance - Corrective, Adaptive, Perfective, Software project management, Risk management, Cost estimation, Software quality assurance.

Reference Books:

1. Software Engineering by Ian Sommerville - Covers SDLC, design, testing, maintenance
2. Software Engineering: A Practitioner's Approach by Roger S. Pressman - Covers testing, design, SRS, quality assurance
3. UML Distilled by Martin Fowler - explanation of UML diagrams, The Art of Software Testing by Glenford J. Myers - Covers unit, integration, system testing