

**CB/CD/CS/
IT225**

SOFTWARE ENGINEERING

**L T P C Int Ext
3 - - 3.0 30 70**

Semester IV [Second Year]

COURSE OBJECTIVES:

The main objectives of this course are:

1. impart knowledge on Principles and practices of process models for software development in software industry.
2. introduce software design concepts
3. Provide awareness on Architectural, Component and User experience design models.
4. Introduce testing techniques and metrics that are applicable to a Software Project.

COURSE OUTCOMES:

After successful completion of the course, the students are able to

1. Classify the Process models, processes and life cycle activities used in all the phases of Software development
2. Apply the software engineering requirements and modeling techniques to solve real- world problems
3. Choose appropriate engineering design model to develop Software components in a software system
4. Specify the software architectural, component level concepts and quality assurance to Software Application
5. Explain the software testing principles, techniques and metrics in the development of a software product

UNIT I

[CO:1] (12)

Software and Software engineering: The Nature of Software- Defining Software, Software Application Domains, Legacy Software, The software Process-The Process frame work, Umbrella activities, Process adaptation.**The Software Process:** Process Models: A Generic Process Model, defining a Framework Activity, identifying a task set, Process Assessment and Improvement, Prescriptive Process Models: The waterfall model, Prototyping Process model, Evolutionary process model, The Unified Process Model.**Agile Development:** What Is Agility? Agility and the cost of change, What Is an Agile Process? Scrum, Other Agile Frameworks- The XP Framework, Kanban, DevOps.

UNIT II

[CO:2,3] (12)

Understanding Requirements: Requirements Engineering, Establishing the Groundwork, Requirements gathering, developing use cases, Building the Analysis Model, Negotiating Requirements, Requirements monitoring, Validating Requirements.**Requirements Modelling:** Requirements Analysis, Scenario-Based Modeling, Class-Based Modeling, Functional Modeling, Behavioural Modeling.**Design Concepts:** Design within the Context of Software Engineering, the Design Process, Design Concepts, the Design Model.

UNIT III

[CO:3,4] (12)

Architectural Design: Software Architecture, Agility and Architecture, Architectural Styles, Architectural Design, Assessing Alternative Architectural Designs, Architectural Reviews.**Modeling Component-Level Design:** What Is a Component? Designing Class-Based Components, Conducting Component Level Design.**Software Quality Assurance** - Elements of Software Quality Assurance, SQA Processes and Product Characteristics, SQA Tasks, Goals, and Metrics, Formal Approaches to SQA, Statistical Software Quality, Assurance, The ISO 9000 Quality Standards, The SQA Plan.

UNIT IV

[CO:5] (12)

Software Testing –Component Level: A Strategic Approach to Software Testing, Planning and Record keeping, Test case design, White box testing, Black-Box-Testing.**Software-Testing Integration**

level: Software Testing Fundamentals, Integration testing, Validation Testing, Testing Patterns.**oftware Metrics and Analytics:** Software Measurement, Software Analytics, Product Metrics, Metrics for Testing, Metrics for maintenance, Process and Project Metrics, Metrics for Quality.

LEARNING RESOURCES:

TEXT BOOK:

Roger Pressman and Bruce Maxim "Software Engineering - A Practitioner's Approach", 9th edition, Tata McGraw-Hill International

REFERENCE BOOK(s):

1. Ian Somerville, Software Engineering. 6 ed, Pearson Education.
2. Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli, Fundamentals of Software Engineering. 2 ed, PHI.
3. Rajib Mall, Fundamentals of Software Engineering. 2nd ed, PHI.