

Course Name: Software Engineering

Course Type: Core

Course Credits: (3+1 credits)

Objectives:

The objective of this course is to understand the Concepts of Software Engineering, Software Development Life Cycle, and to apply project management and analysis principles, and design and testing principles to software project development.

Learning Outcomes:

At the end of this course, students will be able to:

1. Understand the Concepts of Software Engineering.
2. Understand the Software Development Life Cycle.
3. To translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document.
4. To apply project management and analysis principles to software project development.
5. To apply the design & testing principles to software project development.

Unit-I

Introduction to Software Engineering: The Evolving role of Software. Software characteristics and applications, Evolution of Software Engineering, Software crisis and Myths. Software Engineering-A layered Technology, Capability Maturity Model Integration (CMMI). Software Development Process Models (Paradigms): Waterfall Model. Prototyping, Iterative Development, Spiral Model, RAD Model, Incremental Process Model, Evolutionary Process Models, Unified Process, Agile Methodology, SCRUM Approach.

Software process and project metrics, Software Measurement-Size/Function oriented metrics. Metrics for Software Quality.

Unit-II

Software Project: Planning a Software Project. Effort Estimation: (COCOMO and Function Points Model), Project Scheduling, Staffing and Personnel Planning, Software Configuration Management Plan, Quality Assurance Plans, Project Monitoring Plans, Risk Management.

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.

Unit III

Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

System models: Context models, behavioral models, data models, object models, structured methods.

Design Engineering: Design process and design quality, design concepts, the design model.

Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams

Unit IV

Software Quality Assurance, Quality Concepts, SQA Activities, Software Reliability, Measure of Reliability and Availability. The ISO 9000 standards. Software Testing Techniques, Testing fundamentals, White box testing, Basis Path Testing, Control Structure Testing, Black Box testing, validation testing, system testing, the art of debugging, Software Testing strategies. Technical Metrics for Software-Mc-Calls Quality Factors

Recommended Books

- 1) Software Engineering A Practitioner's Approach R Pressman, Mcgraw Hill
- 2) Software Engineering Precise Approach, Pankaj, Wiley
- 3) Fundamentals of Software Engineering, Mall B, Prentice Hall India.
- 4) Software Engineering Udit Aggrawal, SK Kataria