

24CSH303: SOFTWARE ENGINEERING

Hours/Week: 4
Credits: 4

I.A. Marks: 30
Exam. Marks: 70

Course Learning Objectives:

- Be agile software developers with a comprehensive set of skills appropriate to the needs of the dynamic global computing-based society.
 - Capable of team and organizational leadership in computing project settings, and have a broad understanding of ethical application of computing-based solutions to societal and organizational problems.
 - Acquire skills and knowledge to advance their career, including continually upgrading professional, communication, analytic, and technical skills.
 - To understand project scheduling concept and risk management associated to various type of projects.
-

Course Outcomes: After completing the course, the students will be able to,

- Recognize the software engineering and software process.
- Understand different activities of Software process.
- Realize the concepts of agile methods and software testing.
- Learn the techniques of functional and non-functional requirements.
- Familiar with concepts of detailed and object oriented design.
- Define various software application domains and remember different process model used in software development.
- An ability to apply engineering design to produce solutions that meet specified needs.
- Consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors

UNIT-I

12 Hrs.

Introduction: Professional Software Development, Software Engineering Ethics. Case Studies. Software Processes: Models. Process activities. Coping with Change. The Rational Unified Process.

UNIT-II

12 Hrs.

Agile Software Development: Agile methods. Plan-driven and agile development. Extreme programming. Agile project management. Scaling agile methods. Requirements Engineering: Functional and non-functional requirements. The software Requirements Document. Requirements Specification. Requirements Engineering Processes. Requirements Elicitation and Analysis. Requirements validation. Requirements Management, Need for SRS, Characteristics of SRS, organization of SRS document.

UNIT-III

12 Hrs.

Function Oriented Design: Design Principles, Module-Level Concepts, Design Notation and Specification, Structured Design Methodology, Verification, Metrics. Object-Oriented Design: OO Analysis and OO Design, OO Concepts, Design Concepts, Unified Modeling Language (UML), A Design Methodology, Metrics.

UNIT-IV

12 Hrs.

Software Testing: Development testing, Test-driven development, Release testing, User testing. Software Evolution: Evolution processes. Program evolution dynamics. Software maintenance. Legacy system management. Project Planning: Software pricing. Plan-driven development. Project scheduling. Agile planning. Estimation techniques. Quality management: Software quality. Software standards. Reviews and inspections. Software measurement and metrics.

REFERENCE BOOKS:

1. Ian Sommerville, Software Engineering, 9th Edition, Pearson Education, 2012. (Listed topics only from Chapters 1,2,3,4, 5, 7, 8, 9, 24, and 24)
2. Roger S. Pressman, Software Engineering-A Practitioners approach, 7th Edition, Tata McGraw Hill, 2013.
3. PankajJalote, An Integrated Approach to Software Engineering, WileyIndia,2010.

