

MC120-Software Engineering

Objective : This course will be helpful for the student to understand the concept of a software life cycle and the role of process maturity models and apply key elements and common methods for elicitation and analysis to produce a set of software requirements. To distinguish between the different types and levels of testing (unit, integration, systems, and acceptance)

.Learning Outcomes:

After completing the course attendees will be able to:

- appreciate the wider engineering issues that form the background to developing complex, evolving (software-intensive) systems;
- plan a software engineering process to account for quality issues and non-functional requirements;
- Employ a selection of concepts and techniques to complete a small-scale analysis and design project.
- Interact with a client to elicit input, and communicate progress.
- Employ group working skills - including general organization, planning and time management, inter-group negotiation, etc.
- Translate a specification into a design, and then realize that design practically, all using an appropriate software engineering methodology.
- Reflect on the appropriateness of different software engineering methodologies in different circumstances.
- Demonstrate knowledge of the wider software engineering context, software engineering processes and their applicability.

UNIT - I

Introduction to Software Engineering : Software engineering, The software process, Software myths. A generic process model, Process assessment and improvement, Descriptive process models.

UNIT - II

Requirement Engineering and Modelling: Requirement engineering, Building requirement models Requirements analysis, Scenario based modeling, Data modeling concepts, Class based modeling, Flow oriented modeling, Creating a behavioral model

UNIT - III

Design concepts and Architectural design: The design process, design concepts, the design model. Software architecture, Architectural styles, architectural design. What is a component, Designing class based components, conducting component level design, Golden rules, user interface analysis and design.

UNIT - IV

Software Testing: A strategic approach to software testing, test strategies for conventional software, Validation Testing, System Testing.

White box testing, Basis Path Testing, Control Structure Testing, Black Box Testing.

UNIT - V

Product metrics and Quality : Metrics for requirement model, Metrics for design model, Metrics for source code model, Metrics for testing model, Metrics for maintenance model. Introduction to Software quality.

TEXT BOOKS:

1. Roger S. Pressman, "Software Engineering – A practitioner's Approach", Seventh Edition, McGraw-Hill International Edition
2. Ian Sommerville, "Software Engineering", 7th Edition, International Computer Science Series.

REFERENCES BOOKS:

1. K.K. Agarwal & Yogesh Singh, "Software Engineering", New Age International Publishers
2. Waman S Jawadkar, "Software Engineering principles and practice", The McGraw-Hill