3/4 B.Tech - SECOND SEMESTER

IT6T1 **SOFTWARE ENGINEERING** **Credits: 3**

Lecture: 3 Periods/week **Internal assessment: 30 marks**

Practice/Interaction: 1Period/week **Semester end examination: 70 marks**

Objectives:

- To understand Basic Software Engineering principles, practices, and its applications.
- To introduce software process models, identifying software requirements and steps in preparing Software Requirement Specification.
- To understand the project management techniques.
- To create awareness on different software testing approaches and assessing the quality of software.

Outcomes:

Students will be able to

- Acquire knowledge of basic Software engineering principles and its applications.
- Understand the different software process models.
- Prepare effective Software Requirement Specification document.
- Understand the importance of project management.
- Understand different testing approaches and to ensure quality of software.

Syllabus:

Unit - I

Introduction to Software Engineering: The evolving role of software, Changing Nature of Software, Software myths.

The software problem: Cost, schedule and quality, Scale and change .

Unit - II

Software Process: Process and project, component software process, Software development process models: Waterfall model, prototyping, iterative development, relational unified process, time boxing model, Extreme programming and agile process, using process models in a project. Project management process.

Unit - III

Software requirement analysis and specification: Value of good SRS, requirement process, requirement specification, functional specifications with use-cases, other approaches for analysis, validation.

Software Architecture: Role of software architecture, architecture views, components and connector view, architecture styles for C & C view, documenting architecture design, evaluating architectures.

Unit - IV

Planning a software project: Effort estimation, project schedule and staffing, quality planning, risk management planning, project monitoring plan, detailed scheduling.

Design: Design concepts, function-oriented design, object oriented design, detailed design, verification, metrics

Unit - V

Coding and Unit Testing: Programming principles and guidelines, incrementally developing code, managing evolving code, unit testing, code inspection, metrics.

Testing: Testing concepts, testing process, black-box testing, white-box testing, and metrics.

Text Books:

1. Software Engineering, A Precise approach, Pankaj Jalote, Wiley.
2. Software Engineering, 6th Edition, Roger S.Pressman , TMH.

Reference Books:

1. Software Engineering, 8/e, Somerville, Pearson.
2. Software Engineering principles and practice, W S Jawadekar, TMH.
3. Software Engineering concepts, R Fairley, TMH.

e-Learning Resources:

1. http://www.cse.iitd.ac.in/ConciseIntroToSE
2. http://nptel.ac.in/downloads/106105087/