**SOFTWARE ENGINEERING**

**COURSE OUTCOMES**

At the end of the course, the student will develop ability to

1. Define a plan to the software product by adopting suitable process model.
2. Summarize requirements of the software product and produce design modules from analyzed requirements.
3. Create the programs according to programming standards.
4. Apply various testing strategies on the product and evaluate the product performance.
5. Recommend the changes to the system for improvement or Rejuvenation.
6. Illustrate the entire process with the help of documentation

**UNIT I**

**Introduction to Software Engineering**

The evolving role of software

**A Generic view of process:** Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI),

**Process Models**

The waterfall model, Incremental process models, Evolutionary process models, The Unified process. – Agile process.

Product roadmaps, product backlog, ,scrum, sprint backlog, sprint retrospective, backlog grooming, epic, features, users stories etc.

**UNIT II**

**Software Requirements**

Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document.

**Requirements Engineering Process**

Feasibility studies, Requirements elicitation and analysis, Requirements validation.

**UNIT III**

**Unified Modelling Language**

Introduction, Diagrams-Use Case, Class, Collaboration, Sequence, Activity, State chart, Component and Deployment.

Creating an architectural design: Software architecture, Data design, Architectural styles and patterns, Architectural Design.

**UNIT IV**

**Performing User Interface Design**

Golden rules, User interface analysis and design, interface analysis, interface design steps, Design evaluation.

**Testing Strategies**

A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging.

**UNIT V**

Software code analysis, check for vulnerabilities, CWE (common weakness/Vulnerability Enumeration).

Product metrics: Software Quality, Metrics for Analysis Model, Metrics for Design Model, Metrics for source code, Metrics for testing, Metrics for maintenance.

Metrics for Process and Products: Software Measurement, Metrics for software quality.

**UNIT VI**

**Risk Management**

Reactive vs. Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM Plan.

**Quality Management**

Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards.

**TEXT BOOKS**

1. Roger S Pressman, “Software Engineering: A Practitioner's Approach”, 6th Edition, TMH.
2. Ian Sommerville, “Software Engineering” 7th Edition, TMH.

**REFERENCES BOOKS**

1. Shari Lawrence P Fleeger and Joanne M. Atlee, “Software Engineering: Theory and Practice”, 4th Edition, Pearson Education.
2. PedryczWitold and Peters James F, “Software Engineering”, John Wiley.
3. Hans van Vliet, “Software Engineering: Principles and Practice”, 3rd Edition, TMH.

**WEB LINKS**

* + - * 1. http://wps.prenhall.com/esm\_pfleeger\_softengtp\_4/