SOFTWARE ENGINEERING

Instruction 4 Periods per week
Duration of University Examination 3 Hours
University Examination 75 Marks
Sessional 25 Marks

UNIT -I

Introduction to Software Engineering:
Generic view of Process: Software Engineering, Process Framework, CMM, Process Patterns, Process Assessment, Personal and Team Process, Process Technology, Product and process.
Process Models: Perspective Models, Waterfall Model, Incremental Process Models, Evolutionary Process Models, Specialized Process Models, The Unified Process.
An Agile View of Process: What is Agility, Agile Process, and Agile Process Models.

UNIT-II

Planning and Managing the Project: Tracking Progress, Project Personnel, Effort Estimation, Risk Management, the Project Plan, Process Models and Project Management, Information Systems Example, Realtime Example.
Requirement Engineering: A bridge to design and construction, Requirement Engineering tasks, Initiating Requirement Engineering Process, Eliciting Requirement, Developing Uses cases, Building the Analysis Model, Negotiating Requirements, Validating Requirements.

UNIT-III

Building the Analysis Model: Requirements Analysis Modeling approaches, Data modeling concepts, Object oriented analysis , Scenario based modeling, Flow oriented modeling, Class-based modeling, Creating a Behavioral Modeling.
Design Engineering: Design with in the context of SE, Design Process and Design quality, Design concepts, The Design Model, Pattern-based Software Design.

UNIT-IV

Creating Architectural Design: Software architecture, Data design, Architectural Styles and Patterns, Architectural Design, Assessing alternative Architectural Designs, Mapping data flow into software Architecture.
Modeling Component-Level Design: What is a Component, Designing Class-Based components, Conducting Component–level Design, Object Constraint Language, Designing Conventional Components.
Performing User Interface Design: The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, Design Evaluation**.**

UNIT-V

Testing Strategies: A Strategic approach to software testing, strategic issues, test strategies for O-O software, validation testing, system testing, art of debugging.

Testing Tactics: Software Testing Fundamentals, Black-Box and white box Testing, basis path testing, Control Structure Testing, O-O Testing methods, Testing Methods applicable on the class level, inter class Test case design, Testing for Specialized environments, architectures and applications, Testing Patterns.

Product Metrics: Software quality, A framework for product metrics, Metrics for the analysis model, metrics for the Design model, metrics for source code, Metrics for Testing, Metrics for maintenance.

Suggested Reading:

1. Roger S. Pressman, *"Software Engineering –A Practitioners Approach",* 6th Edition, Pearson Education, India, 2005.
2. Shari Lawrence Pfleeger, *"Software Engineering Theory and Practices"* 4th Edition - Pearson Education, India, 2011.
3. Pankaj Jalote, *"An Integrated Approach to Software*