L	T	P	C
3	0	0	3

Course Code: CSE314R01

SOFTWARE ENGINEERING PRACTICES

Course Objective:

This course will help the learner to design a real world project in a systematic manner and also assess the possible problems involved during System Design

UNIT - I **11 Periods**

Software Engineering: The Nature of Software - Software Process - Engineering Practice - A Generic Process Model - Defining a Framework Activity - Identifying a Task Set - Process Patterns - Process Assessment and Improvement. **Process Models**: Prescriptive Process Models - Specialized Process Models - The Unified Process - Personal and Team Process Models. **Agile Model**: Agility and Cost of Change - Agile Process - Extreme Programming - Other Agile Process Models - CMMI.
Case Study*: Agile Manifesto, Pass on the Secret Session, Waterfall model Session.

UNIT - II **11 Periods**

Requirement Engineering:Requirement Engineering - Establishing the Groundwork - Eliciting Requirements - Developing Usecases - Building the Analysis Model - Negotiating and Validating Requirements. **Software Project Estimation**: Project Planning Process - Resources - Decomposition Techniques - Empirical Estimation Models - Estimation for OO Projects -**Requirement Modeling**: Scenario-based methods - class-based methods -Creating a Behavioral Model - Identifying Events with the Use Case - State Representations.
Case Study*:Cause Effect Session, Project Inception, Customer Collaboration, Speed Boat Session

UNIT - III **11 Periods**

Design Engineering: Design within the Context of Software Engineering - The Design Process -Design Concepts - The Design Model.**Architectural Design**: Software Architecture - Architectural Genres - Architectural Styles - Architectural Considerations - Architectural Decisions - Representing System in Context - Defining Archetypes - Refining the Architecture into Components - Describing the Instantiations of the System.
Component Design:Traditional, Object-oriented and Process related View of Components - Designing Class-based Components - Conducting Component-Level Design - Component-based Development. User **Interface Design**: The Golden Rules - User Interface Analysis and Design - Interface Analysis & Design Steps - Design Evaluation.

UNIT - IV **12 Periods**

Testing: Testing Strategies - Strategic Approach - Strategic Issues - Test Strategies for Conventional and Object-oriented Software – Validation Testing – System Testing – Art of Debugging. **Conventional Testing:** White-Box Testing - Basis Path Testing - Control Structure Testing -Black-Box Testing –Model-based Testing – Testing for Real-Time

Systems. **Object-oriented Testing:** Testing OOA and OOD Models - OO Testing Strategies – OOTesting Methods -Testing Methods Applicable at Class Level - Interclass Test Case Design.
***Case studies are meant only for tutorial sessions**

TEXT BOOK

1. Roger S, Pressman and Bruce R, Maxim. *Software Engineering A Practitioner's Approach*, McGraw Hill, Eighth Edition, 2015.

REFERENCES

1. Pankaj Jalote. *An Integrated Approach to Software Engineering*, Narosa Publishing House, Third Edition, 2014.
2. Rajib Mall. *Fundamentals of Software Engineering*, Prentice Hall of India, Third Edition, 2009.

ONLINE MATERIALS

1. http://nptel.ac.in/courses/106101061/
2. http://nptel.ac.in/courses/Webcoursecontents/IIT%20Kharagpur/Soft%20Engg/New_index1.html

UNITWISE LEARNING OUTCOMES

Upon successful completion of each unit, the learner will be able to

Unit I	• Build software projects using different process models • Implement agile software process model
Unit II	• Dramatize the activities involved in analysing the requirements • Estimate LOC and FP based efforts in person/months • Design usecase, data, class and behavioural models for the software
Unit III	• Illustrate architectural and component models • Create effective user interfaces
Unit IV	• Test software code based on functional and non-functional aspects of the software • Prepare test plans and strategies for conventional and OO software

COURSE LEARNING OUTCOMES

Upon successful completion of this course, the learner will be able to

- Select an appropriate process model for software product requests
- Implement agile software process models for a given scenario
- Identify the stakeholder's requirements and employ the models for analysing there requirement
- Estimate software cost and efforts in person/months
- Construct Use cases, Activity, Collaboration and state transition diagrams that maps software requirements
- Design architecture for developing a project based on user requirements
- Create test plans and employ testing strategies for verifying functional and non-functional requirements of the project