SOFTWARE ENGINEERING AND DESIGN

Regulation	Year/ Sem	Course Code	Category	Periods / Week			Credits	Maximum Marks		
				L	T	P	C	CIA	SEE	Total
R20A	II – I	A53E1	PC	3	-	-	3	30	70	100

Prerequisites:

Computational Thinking and Problem Solving Techniques.

Course Objectives:

1. To define software engineering process and practices, and demonstrate various process models
2. To develop software requirements and the SRS document.
3. To make use of structural UML diagrams.
4. To make use of structural UML diagrams.
5. To learn levels of software testing and Methodologies.

Syllabus:

UNIT–I: (10 Periods)

Software Process: Process and project, Software development process models - Waterfall model, relational unified process, Time boxing model, Agile Manifesto, Extreme programming Model, Using process models in a project, Project management process.

UNIT–II: (10 Periods)

Functional Requirements - Non functional Requirements - Software requirement analysis and specification: Value of good SRS, requirement process, requirement specification using Data flow Diagram. Case study - ATM, Library Management System.

UNIT–III: (9 Periods)

Software Design - Introduction to UML - Structural UML diagrams - Class Diagram, Package Diagram, Component Diagram, and Deployment Diagram. Case studies - Point of Scale System, Online Ticket Reservation System.

UNIT–IV: (9 Periods)

Behavioral UML Diagrams - Usecase diagram, Interaction Diagram, Activity diagram, State Machine diagram. Case studies - Point of Scale System, Online Ticket Reservation System.

UNIT –V: (10 Periods)

Software Testing: Testing Concepts, Levels of Testing, Testing Process, Black-Box Testing, White- Box Testing Techniques, Art of Debugging.

Course Outcomes: Upon completion of the course, the students will be able to

1. Students will be able to choose appropriate process model depending on the user requirements. (BL- 1)
2. Demonstrate the principles and requirements at various phases of software development.(BL- 2)
3. Model structural UML Diagrams. (BL- 3)
4. Model behavioral UML Diagrams. (BL- 3)
5. Distinguish different testing strategies. (BL- 3)

Text Books :

1. A Concise Introduction to Software Engineering (Undergraduate Topics in Computer Science), Pankaj Jalote, Springer International Edition. (Unit 1, 2, &5)
2. Craig Larman," Applying UML and Patterns: An Introduction to object-oriented Analysis and Design and iterative development", Third Edition, Pearson Education, 2005 (Unit 3&4)

References :

1. Roger S, "Software Engineering – A Practitioner's Approach", seventh edition, Pressman, 2010.
2. earson Edu, "Software Engineering by Ian sommerville", 9 th edition, 2010.
3. Mike O'Docherty, "Object-Oriented Analysis & Design: Understanding System Development with UML 2.0", John Wiley & Sons, 2005.