SEMESTER VI

23CSE311   **Software Engineering**   **L-T-P-C: 3-0-2-4**

Course Objectives

● This course addresses issues in the engineering of software systems and development using live case studies from industries.

● The objectives of this course are to introduce basic software engineering concepts; to introduce the Agile Software development process; hands-on training (experiential learning) using state-of-the-art tools to understand the concepts learnt in the class.

● The course helps students to be industry-ready in terms processes, tools and terminologies from agile and devops point of view

Course Outcomes

CO1: Understand process models and apply Agile methodologies for proficient software process management.

CO2: Apply requirement engineering principles to analyze, model, and validate requirements for effective software solutions.

CO3: Design and implement robust software architectures and intuitive user interfaces, following industry best practices.

CO4: Implement comprehensive testing strategies to ensure high software quality.

CO5: Utilize industry-standard tools and Scrum for efficient software project management and collaboration.

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2										1		1	
CO2	2	3	2										2	1
CO3	3	1	3		3				3	3	1		2	1
CO4	2	2	3		3				3	3	1		2	1
CO5	2		3		3				3	3	1	2	3	2

CO-PO Mapping

Syllabus

Unit 1

Process Models – overview, Introduction to Agile, Agile Manifesto, principles of agile manifesto, over-view of Various Agile methodologies - Scrum, XP, Lean, and Kanban, Agile Requirements - User personas, story mapping, user stories, estimating and prioritizing stories, INVEST, acceptance criteria, Definition of Done, Release planning Key aspects of Scrum: roles - Product Owner, Scrum Master, Team, Manager in scrum

and product backlog Scrum process flow: product backlog, sprints backlog, scrum meetings, demos. How sprint works: Sprint Planning, Daily scrum meeting, updating sprint backlog, Burn down chart, sprint review, sprint retrospective. Scrum Metrics-velocity, burn down, defects carried over.

Unit 2

Traditional process Models: Waterfall, incremental, evolutionary, concurrent. Requirements Engineering: Tasks Initiation- Elicitation-Developing Use Cases-Building the analysis Model-Negotiation- Validation Requirements Modelling - building the analysis model, Scenario based methods, UML Models, Data Models. Design engineering Design concepts, Design models, software architecture, architectural styles and patterns, Architectural design: styles and patterns, architectural design, Refining architecture to components. Performing user interface Design-Golden Rules - User Interface Analysis and Design- Interface Analysis-Interface design steps.

Unit 3

Testing strategies and tactics: Unit testing, integration testing, validation and system testing, Devops.

Textbook(s)

Pressman R S, Bruce R.Maxim, "Software Engineering - A Practitioner's Approach". Eighth Edition, McGraw-Hill Education, 2019.

Reference(s)

Crowder JA, Friess S. "Agile project management: managing for success". Cham: Springer International Publishing; 2015.

Stellman A, Greene J. "Learning agile: Understanding scrum, XP, lean, and kanban". O'Reilly Media, Inc.; 2015.

Gregory J, Crispin L. "More agile testing: learning journeys for the whole team". Addison-Wesley Professional; 2015.

Rubin KS. "Essential Scrum: a practical guide to the most popular agile process". Addison-Wesley; 2012.

Cohn M. "User stories applied: For agile software development". Addison-Wesley Professional; 2004.

Evaluation Pattern: 70:30

Assessment	Internal	External
Midterm	20	
*Continuous Assessment (Theory) (CAT)	10	
*Continuous Assessment (Lab) (CAL)	40	
**End Semester		30 (50 Marks; 2 hours exam)

*CAT includes Quizzes and Tutorials
*CAL – Can be Sprint Reviews, Lab Assessments, Project, Case Study and Report
**End Semester can be theory examination/ lab-based examination/ project presentation