SEMESTER VI

19CSE314	SOFTWARE ENGINEERING	L-T-P-C: 2-0-3-3

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

CO 1: Understand and apply the principles of software engineering
CO 2: Understand various software process models
CO 3: Apply the appropriate software design methodology for a given scenario
CO 4: Evaluate a system developed for real-world applications in Agile Mode
CO 5: Understand and implement various industry standards

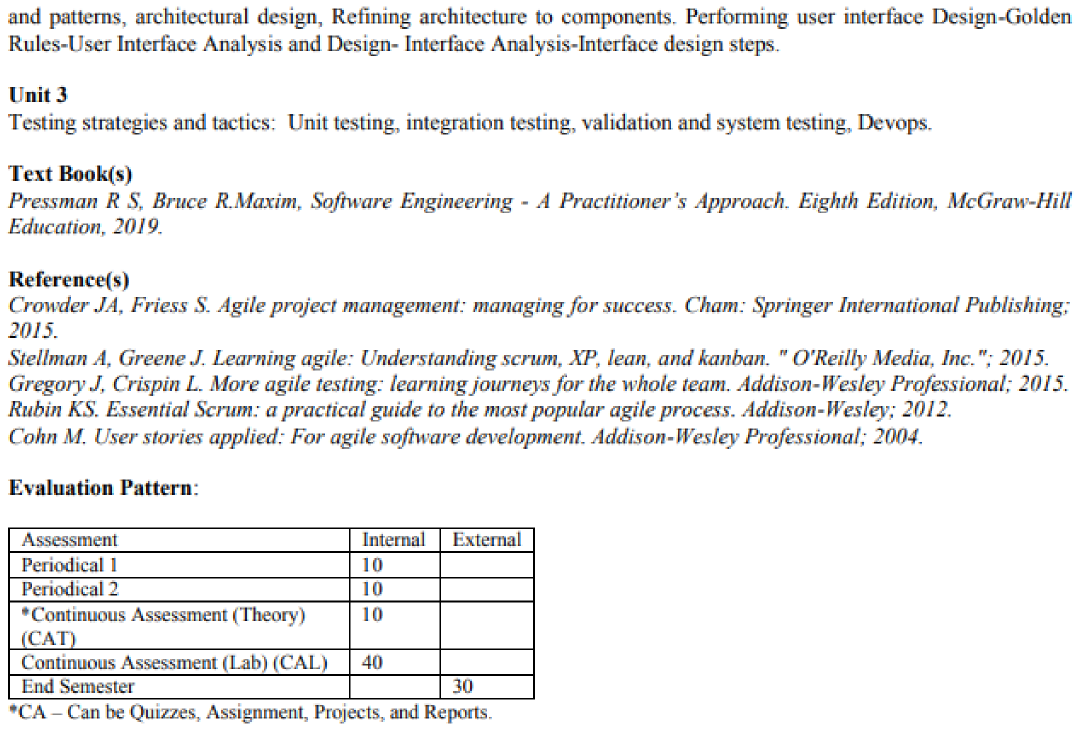
CO-PO Mapping

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

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.

Text Book(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:

Assessment	Internal	External
Periodical 1	10	
Periodical 2	10	
*Continuous Assessment (Theory) (CAT)	10	
Continuous Assessment (Lab) (CAL)	40	
End Semester		30

*CA – Can be Quizzes, Assignment, Projects, and Reports.

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

EVALUATION PROCEDURE

Internal: 70

External: 30

Mid Term Online	10 marks
Mid Term Viva	10 marks
End Semester (Online exam+Review+viva)	**30 marks**
Continuous Assessment (Theory) 3 common quizzes	10 marks (3+3+4)
Sprint # 1 (Project)	20 marks
Sprint # 2 (Project)	20 marks
TOTAL	**100 Marks**