



MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

COURSE CONTENT

| AGILE METHODOLOGIES | | | | | | | | |
|--|-----------------------|------------------------|---|---|-------------------|---------------|-----|-------|
| I Semester: CSE | | | | | | | | |
| Course Code | Category | Hours / Week | | | Credits | Maximum Marks | | |
| 2515827 | Foundation | L | T | P | C | CIA | SEE | Total |
| | | 3 | 0 | 0 | 3 | 40 | 60 | 100 |
| Contact Classes: 45 | Tutorial Classes: Nil | Practical Classes: Nil | | | Total Classes: 45 | | | |
| Prerequisites: A course on "AGILE METHODOLOGIES" | | | | | | | | |

Course Overview:

This course introduces the principles and values of Agile development based on the Agile Manifesto, emphasizing iterative, incremental, and customer-centric software delivery. It covers popular Agile frameworks such as Scrum and Kanban, along with roles, artifacts, and ceremonies.

Course Objectives:

1. To understand the principles and practices of Agile software development and Extreme Programming (XP).
2. To develop skills in Agile collaboration, teamwork, and customer-centric development practices.
3. To apply Agile planning, estimation, and risk management techniques in software projects.
4. To implement continuous integration, version control, and iterative release practices for quality software delivery.
5. To use Agile development techniques such as test-driven development, refactoring, and exploratory testing for maintainable software systems.

Course Outcomes: After Completion of the Course, Students should be able to

1. Apply Agile and Extreme Programming principles, including XP lifecycle and team practices, to improve software project adaptability and delivery efficiency.
2. Implement collaborative techniques such as pair programming, continuous integration, and iteration demos to enhance team productivity in software development projects.
3. Analyze release management processes, including version control, fast builds, and documentation practices, to ensure high-quality software delivery.
4. Design effective planning strategies, including risk management, iteration planning, and story estimation, for predictable and efficient software development.
5. Evaluate development practices such as test-driven development, refactoring, and exploratory testing to optimize software performance and maintainability.

UNIT - I:

Introduction Extreme Programming (XP) - Agile Development
Why Agile - Understanding Success, Beyond Deadlines, Importance of Organizational Success, Introduction to Agility How to Be Agile - Agile methods, don't make your own method, Road to mastery Understanding XP (Extreme Programming) - XP life cycle, XP team, XP Concepts Adopting XP - Knowing whether XP is suitable, Implementing XP, assessing Agility Practicing XP - Thinking - Pair Programming, Energized work, Informative Workspace, Root cause Analysis, Retrospectives

UNIT - II:

Collaborating
Trust, Sit together, Real customer involvement, Ubiquitous language, meetings, coding standards, Iteration demo, Reporting

UNIT - III:

Releasing
Bugfree Release, Version Control, fast build, continuous integration, Collective ownership, Documentation

UNIT - IV:

Planning
Version, Release Plan, Risk Management, Iteration Planning, Slack, Stories, Estimating

UNIT - V:

Developing:
Incremental requirements, Customer tests, Test driven development, Refactoring, Incremental design and architecture, spike solutions, Performance optimization, Exploratory testing

TEXT BOOKS:

1. The art of Agile Development, James Shore and Shane Warden, 11th Indian Reprint, O'Reilly, 2018.

REFERENCE BOOKS:

1. Learning Agile, Andrew Stellman and Jennifer Greene, O'Reilly, 4th Indian Reprint, 2018.
2. Practices of an Agile Developer, Venkat Subramaniam and Andy Hunt, SPD, 5th Indian Reprint, 2015.
3. Agile Project Management - Jim Highsmith, Pearson Low price Edition 2004.

ELECTRONIC RESOURCES:

1. <https://www.guru99.com/agile-scrum.html>
2. <https://www.softwaretestinghelp.com/agile-testing-tutorials/>
3. <https://www.tutorialspoint.com/agile/index.htm>
4. <https://www.edx.org/learn/agile-development>

MATERIALS ONLINE:

1. Course template
2. Tutorial question bank
3. Tech talk and Concept Video topics
4. Open-ended experiments
5. Definitions and terminology
6. Assignments
7. Model question paper – I
8. Model question paper – II
9. Lecture notes
10. E-Learning Readiness Videos (ELRV)

