Pokhara University
Faculty of Science and Technology

Course Code.: CMP 376 (3 Credits) Full marks: 100 Course Title: Agile Software Development **(3-1-2)** Pass marks: 45 Nature of the course: Theory/Practice/Theory & Practice Time per period: 1 hour Year, Semester: III/VI Total periods: 45 Level: Bachelor
Program: Bachelor of Software Engineering

1. Course Description

This course provides a comprehensive understanding of Agile methodologies and their application in software development. Students will learn the core principles and values of Agile, gain hands-on experience with several frameworks and Agile tools for project management and collaboration, and manage changing project requirements. Emphasis is placed on fostering effective communication and teamwork within Agile teams and implementing Agile practices in real-world scenarios.

2. General Objectives

By the end of this course, students will be able to:

- Understand the core principles and values of Agile methodologies.
- Apply various Agile frameworks such as Scrum, Kanban, and Extreme Programming (XP).
- Utilize Agile tools for project management and collaboration.
- Manage and adapt to changes in project requirements.
- Foster effective communication and teamwork within Agile teams.
- Implement Agile practices in real-world software development scenarios.

3. Contents in Detail

Specific Objectives	Contents
• Understand the manifesto and importance of Agile Software Development with respect to different nature of projects and development environment.	1.2. Agile Manifesto and Principles - Understanding the values and principles that define Agile 1.3. Benefits and Challenges of Agile - Pros and cons of adopting Agile in different environments
• Relate the knowledge of various agile frameworks and several events under them.	**Unit 2. Agile Frameworks (7 hours)** 2.1. Scrum Framework (2 hours) - Roles: Scrum Master, Product Owner, Development Team, Artifacts: Product Backlog, Sprint Backlog, Increment, Events: Daily Stand-up, Backlog refinement/grooming, Sprint Planning, Sprint Review, Sprint Retrospective 2.2. Kanban (1 hours) - Kanban principles, Visualizing workflow, Managing work in progress (WIP) 2.3. Extreme Programming (XP) (2 hours) - Core practices: Test-Driven Development, Pair Programming, Continuous Integration, XP roles and values 2.4. Lean Software Development (2 hours) - Lean principles, Eliminating waste, Continuous improvement
• Understand how to estimate the agile projects, manage and monitor the development progress via various charts and dashboards. **Unit 1: Introduction to Agile Software Development (3 hours)** 1.1. History of Software Development Methodologies - Overview of traditional vs. Agile methodologies	**Unit 3: Agile Project Management (6 hours)** 3.1. Agile Planning and Estimation - User stories, Story points and planning poker, Release planning 3.2. Managing Agile Projects - Iterative development, Incremental delivery, Tracking progress with burndown/burnup charts/dashboards. 3.3. Risk Management in Agile - Identifying and mitigating risks, Adapting to changes

• Adept in using various Agile tools	and technologies for real-world

projects	Features and best practices 4.2. Collaboration Tools, Enhancing team communication 4.3. Continuous Integration/Continuous Deployment (CI/CD) – Tools and Implementing CI/CD pipelines
• Able to understand different agile practices and techniques, code quality, team collaboration, and codebase maintainability.	**Unit 5: Agile Practices and Techniques (6 hours)** 5.1. Test-Driven Development (TDD) - Writing and executing tests before code 5.2. Behavior-Driven Development (BDD) - Bridging the gap between business and technical teams 5.3. Pair Programming and Mob Programming - Enhancing code quality and team collaboration 5.4. Refactoring and Code Reviews - Improving codebase maintainability
• Explore the SAFe framework, the Scrum of Scrums approach for managing and coordinating large scale Agile projects across multiple teams.	**Unit 6: Scaling Agile (4 hours)** 6.1. Scaling Frameworks - SAFe (Scaled Agile Framework) 6.2. Managing Large Agile Teams - Coordination and communication strategies, Integrating multiple Agile teams
• Understand several Agile KPIs like velocity, cycle time, and lead time and interpret from various reporting tools and dashboards for visualizing and continuous improvement.	**Unit 7: Agile Metrics and Reporting (3 hours)** 7.1. Key Performance Indicators (KPIs) Velocity, cycle time, lead time 7.2. Reporting Tools and Dashboards Visualizing team performance 7.3. Continuous Improvement through Metrics Using data to drive process enhancements **Unit 8: Agile in Practice (6 hours)**
• Analyze case studies of Agile implementations in real-world examples, applying the methodologies in a group team project, present and evaluate to demonstrate practical skills.	8.1. Case Studies of Agile Success and Failures Analyzing real-world examples 8.2. Agile Transformation - Steps to transition from traditional to Agile 8.3. Assignment/ Project - Applying Agile methodologies in a team project, Presentations and evaluations
• Learn how to communicate effectively, handle conflicts, lead with a servant leadership approach,	**Unit 9: Soft Skills for Agile Teams (4 hours)**
Unit 4: Agile Tools and Technologies (6 hours) 4.1. Project Management Tools,	9.1. Effective Communication - Facilitating open and honest dialogue

and create a team culture of trust and collaboration.	and resolving team conflicts 9.3. Leadership in Agile Teams - Servant Leadership Principles 9.4. Building a Collaborative Culture - Fostering trust and accountability

4. Methods of Instruction •

Lectures:

9.2. Conflict Resolution - Managing

a) Lectures will cover core theoretical concepts and provide foundational knowledge on Agile Software Development. They will be interactive, using multimedia presentations, real-world examples, project works, group discussions, and case studies to illustrate key points. Lectures will encourage questions and discussions to deepen understanding.

- **Group Projects and Collaborative Learning:**
  a) Students should find a new problem/case studies/scenarios and work on the Agile Software Development. They will work on each of the topics studied in the lecture, implement the concept learned in the group project, and present the project work progressively. The project work can be done in groups with at most three members in each group.
  b) Case studies will be used to demonstrate how Agile methodologies are implemented in real-world projects. Students will analyze these cases, identify challenges, and propose solutions based on various nature of projects.
  c) Best-of-breed Tools are to be used for Project management, monitoring and control, Collaboration and CI/CD Development
    - ▪ Project management, monitoring and control - Jira, Trello, Asana, Monday.com, Click Up, etc
    - ▪ Collaboration - Slack, Microsoft Teams, Confluence, Google Suite
    - ▪ CI/CD Development - Jenkins, GitLab

- **Guest Lectures and Industry Insights:**
  a) Industry experts can be invited to conduct workshops, providing real-world industry insights.

5. Practical/project Works

Project work can be done in groups with at most three members in each group. The learned concepts can be implemented in projects and can be presented progressively.

6. Evaluation system and Students' Responsibilities

Evaluation System

In addition to the formal exam(s) conducted by the Office of the Controller of Examination of Pokhara University, the internal evaluation of a student may consist of class attendance, class

participation, quizzes, assignments, presentations, written exams, etc. The tabular presentation of the evaluation system is as follows.

Internal Evaluation Weight Marks External Evaluation Marks
Theory 30 Attendance & Class
Participation 10% Assignment 20%
Internal Assessment 70% **Practical**
(Project Work) 20 Progressive Project
work 40% Project Presentation 30% Final
Project Report 30% **Total Internal** 50

Semester End Examination

Full Marks: 50 + 50 = 100

Students' Responsibilities:

Each student must secure at least 45% marks in the internal evaluation with 80% attendance in the class to appear in the Semester End Examination. Failing to obtain such a score will be given NOT QUALIFIED (NQ) and the student will not be eligible to appear in the End-Term examinations. Students are advised to attend all the classes and complete all the assignments within the specified time period. If a student does not attend the class(es), it is his/her sole responsibility to cover the topic(s) taught during the period. If a student fails to attend a formal exam, quiz, test, etc. there won't be any provision for a re-exam.

7. Prescribed Books and References

Text Book

- Robert C. Martin and Micah Martin, Agile Software Development: Principles, Patterns, and Practices, Prentice Hall, Latest edition

Reference Books

- Jeff Sutherland, The Art of Doing Twice the Work in Half the Time, Crown Business
- David J. Anderson, Kanban: Successful Evolutionary Change for Your Technology Business, Blue Hole Press
- Kent Beck and Cynthia Andres, Extreme Programming Explained: Embrace Change, Addison-Wesley