

Devi Ahilya Vishwavidhyalaya, Indore, India Institute of Engineering & Technology				II Year B.Tech. (Computer Science and Engineering)		
Course Code & Name	Instructions Hours per Semester and Credits					
3RCPC3 Software Engineering and Agile Development	Classroom Instruction (CI)		Lab Instruction (LI)	Term Work (TW) and Self Learning (SL)	Total no. of Hours Per semester	Total Credits (Total Hours/30)
	L	T	P	TW+SL	120	4
	30	10	20	60		

Prerequisite: Knowledge of a programming language, preferably object oriented and mid-size project work.

Learning Objectives:

1. To introduce the various aspects of Software Engineering applied for the software development.
2. Compare and contrast different software development methodologies and identify appropriate contexts for their use.
3. Learning techniques for gathering and analysing software requirements, including stakeholder interviews and requirements documentation.
4. Identify and apply key design principles and architectural patterns in software design.
5. Develop and execute test plans, including unit testing, integration testing, and system testing, to ensure software quality.

COURSE CONTENTS

Unit 1: Introduction to Software Engineering and Agile Principles

Nature and scope of software engineering, Software development life cycle (SDLC) models: Waterfall, Spiral, RAD, V-Model, Agile Manifesto: values and principles, Comparison of traditional vs. Agile methodologies, Overview of Agile frameworks: Scrum, XP, Kanban, Lean

Unit 2: Requirements Engineering and Agile Planning

Types of Requirements, Functional and non-functional, Requirement elicitation techniques, interviews, prototyping, brainstorming, Software Requirements Specification (SRS). Agile requirements: user stories, epics, story mapping, Agile estimation: planning poker, velocity, release planning

Unit 3: Software Design and Agile Development Practices

Design principles, modularity, abstraction, cohesion, coupling, UML diagrams, class, sequence, activity, use-case, Agile design practices: refactoring, pair programming, simple design, Continuous integration and version control (Git, GitHub), Agile tools: JIRA, Trello, GitLab

Unit 4: Testing, Quality Assurance, and Continuous Delivery

Software testing levels: unit, integration, system, acceptance testing, black-box vs. white- box

testing, Agile testing: TDD, BDD, exploratory testing, Test automation and CI/CD pipelines, Software quality metrics and SQA activities

Unit 5: Maintenance, Dependable systems and Scaling

Software maintenance corrective, adaptive, perfective, Legacy Systems, Complexity metrics, Dependability, Dependability Properties, Dependable Process and agility.

Course Outcome:

CO. No.	CO
CO1.	Students will be able to choose appropriate development models (e.g., Waterfall, Scrum) based on project needs and constraints.
CO2.	Learners will demonstrate the ability to create scalable, maintainable designs using object-oriented and Agile design techniques
CO3.	Participants will show proficiency in eliciting, documenting, and evolving requirements through SRS documentation and Agile artifacts like user stories.
CO4	Students will apply testing techniques from unit to system level, and learn CI/CD pipelines for ongoing deployment and feedback.
CO5	Learners will be equipped to lead Agile teams, use tools like JIRA/Trello, perform cost estimation, and manage risk across the software lifecycle.

BOOKS RECOMMENDED:

- [1] C.Gezzi, M. Jazayeri and D. Mandrioli, *Fundament of software Engineering*, 2 nd edition Pearson, 2002
- [2] R.S. Pressman, *software Engineering A Practitioner Approach*, 7th edition, Mc Graw Hill International Edition 2009
- [3] P.Jalote, Pankaj Jalote's *Software Engineering: A Precise Approach*, Wiley, 2010
- [4] Ian Sommerville, *Software Engineering*, Pearson education, 10th edition, 2015
- [5] Craig Larman, “Agile and Iterative Development: A Manager’s Guide”, Pearson Education, Second Impression, 2007.
- [6] Ken Schwaber, “Agile Project Management with Scrum”, Illustrated, Revised Edition Microsoft Press, 2004.
- [7] KonnorCluster, “Agile Project Management: Learn How To Manage a Project With Agile Methods, Scrum, Kanban and Extreme Programming”, Independently Published, First Edition, 2019.

CO-PO-PSO Relationships

CO	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
3RCPC3.CO1	2	2	-	-	-	-	-	-	-	-	-	3	2	1
3RCPC3.CO2	2	2	3	-	-	3	-	-	-	-	-	3	3	2
3RCPC3.CO3	-	-	-	2	3	3	2	3	-	-	-	3	2	1
3RCPC3.CO4	-	-	3	3	3	-	-	-	3	-	-	3	2	2
3RCPC3.CO5	-	-	-	-	-	2	2	3	-	-	-	3	2	2

List of Practical Assignments in Software Engineering

1. Assignments for identifying category-specific concerns and ethical obligations per the IEEE Code.
2. Assignments to analyze a case study for a safety-critical medical device vs. a social media startup. Select and justify a process model (Waterfall vs. Agile) for each.
3. Assignments to translate high-level customer "wants" into a formal Software Requirements Specification (SRS) document, including functional and non-functional requirements.
4. Assignment to model Use Case diagrams, Class diagrams, and Sequence diagrams to model behavior and structure.
5. Assignment for the Legacy Code Rescue, by perform **Reverse Engineering** to create its design schema.
6. Assignment for **Refactoring** to improve code readability and maintainability without changing its external behavior.
7. Assignment for writing user stories, Backlog & Story Pointing.
8. Assignment to Use Planning Poker to estimate effort and define the "Definition of Done" for a project increment.
9. Assignment to Write Unit Tests for the refactored code, Document bugs found through integration testing in a formal Defect Report.
10. Assignment to write Test Case for login, registration, online shopping item tracker, online shopping wishlist.
11. Assignment for detecting cohesion and coupling for a class design