

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

New Scheme Based On AICTE Flexible Curricula

Artificial Intelligence and Data Science, IV-Semester

AD403: Software Engineering with Agile Methodology

Rationale:

1. The purpose of this subject is to cover the underlying concepts and techniques used in Software Engineering. Some of these techniques can be used in software design & its implementation.

2. To understand the modern way of Software development using Agile methodology.

Prerequisite: - The students should have at least one year of experience in programming a high-level language and databases. In addition, a familiarity with the software development life cycle will be useful in studying this subject.

Course Outcomes: After completing the course student should be able to:

1. Describe the Fundamentals of software Engineering.
2. To Understand Software Development Life Cycle phases and their Role in Software Development.
3. Understand the Software development using Agile methodology.
4. Understand the implementation principles and guidelines for software development using Agile methodology
5. Use implementation techniques of Software architecture for effective software development.
6. Apply core values and principles of Agile for enterprise application development

Unit I: Introduction to Software Engineering

Software Development Life Cycles, SDLC Models: Waterfall, V-Model, Prototype Model, Incremental, Evolutionary, RAD, Spiral. Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Requirements Gathering and Analysis, Software Requirements Specification (SRS). Software Product and Process Characteristics, Software Process Models, Evolutionary Process Models and Agile processes. Software Process customization and improvement, CMM, Product and Process Metrics, Functional and Non-functional requirements, Requirement Sources and Elicitation Techniques, Analysis Modeling for Function-oriented and Object-oriented software development, Use case Modeling, System and Software Requirement Specifications, Requirement Validation, Traceability

Unit II: Software Design, Analysis and Testing

The Software Design Process, Design Concepts and Principles, Software Modeling and UML, Architectural Design, Architectural Views and Styles, User Interface Design, Function oriented Design, SA/SD Component Based Design, Design Metrics. Software Static and Dynamic analysis, Code inspections, Software Testing, Fundamentals, Software Test Process, Testing Levels, Test Criteria, Test Case Design, Test Oracles, Test Techniques, Testing Frameworks, Test Plan, Test Metrics, Testing Tools.

Unit-III: Software Maintenance & Software Project Measurement

Need and Types of Maintenance, Software Configuration Management (SCM), Software Change Management, Version Control, Change control and Reporting, Program

Comprehension Techniques, Re-engineering, Reverse Engineering, Tool Support. Project Management Concepts, Feasibility Analysis, Project and Process Planning, Resources Allocations, Software efforts, Schedule, and Cost estimations, Project Scheduling and Tracking, Risk Assessment and Mitigation, Software Quality Assurance (SQA). Project Plan, Project Metrics.

Unit-IV: Fundamentals of Agile Methodology

Introduction to Agile software development methodology, Life Cycle of Agile development, Agile v/s Traditional software development(Waterfall model)Agile Manifesto: Principles, Benefits and Challenges of Agile, Agile Values, Agile Model, Phases of Agile Model.

Unit-V: Software Development using Agile Methodology

Gathering requirement using Agile way, User Stories: The currency of agile development, Characteristics of good user stories, Generating User Stories, Agile estimation and planning, Implementation of agile, Applying an Agile Mindset to a Project, Roles in agile development, Agile Frameworks: Scrum, Kanban, Crystal, XP, ASD, DSDM.

Practical and Lab work: Lab work should include a running case study problem for which different deliverables set at the end of each phase of a software development life cycle are to be developed. This will include modeling the requirements, analysis, detailed design, implementation, testing, deployment, and maintenance. Subsequently the design models will be coded and tested. For modeling, Open Source tools like StarUML and Licensed Tools like Rational Rose products. For coding and testing, IDE like Eclipse, Net Beans, and Visual Studio can be used.

Text Books:

1. Pankaj Jalote ,”An Integrated Approach to Software Engineering”, Narosa Pub, 2005
2. Rajib Mall, “Fundamentals of Software Engineering” Second Edition, PHI Learning
3. James Shore and Shane Warden, “The Art of Agile Development 1st Edition”, o'reilly books.
4. Dikel, David, D. Kane, and J. Wilson, “Software Architecture: Organizational Principles and Practices”, Prentice-Hall.
5. Mike Cohn, “Agile Estimating and Planning, 1st (first) edition”, Prentice-Hall.

References:

1. R S. Pressman ,”Software Engineering: A Practitioner's Approach”, Sixth edition, 2006, McGraw-Hill.
2. Sommerville,”Software Engineering”, Pearson Education.
3. Richard H.Thayer,”Software Engineering & Project Managements”, Wiley India.
4. Waman S.Jawadekar,”Software Engineering”, TMH.
5. Bob Hughes, M.Cotterell, Rajib Mall “Software Project Management”, McGraw Hill.
6. Bennett, Douglas, “Designing Hard Software: The Essential Tasks”, Prentice-Hall, 1997.
7. The Deadline: A Novel about Project Management, Dorset House