Progressive Education Society's
Modern College of Engineering, **Pune-05.**
Department of Computer Engineering

Class-F.Y.MTech. Computer Engg

Sub: Software Engineering and Project Management (Course Code: CSE01502)

Question bank

Unit 1

1. Apply the principles of service-oriented architecture (SOA) to design a healthcare management system.
2. Compare formal specifications and informal specifications in terms of precision, clarity, and implementation challenges.
3. Examine the qualities of a good specification document and analyze how they contribute to successful project outcomes.
4. Analyze the challenges of requirements classification in large-scale systems and suggest strategies to address them.
5. Illustrate the life cycle of the unified process model.
6. Use Petri nets to represent the process flow in a ticket booking system.
7. Draw a deployment diagram to showcase the hardware and software architecture of a distributed system.
8. State the main focus of the unified process model
9. Show how validation techniques ensure correctness in software requirements
10. Apply the Unified Modeling Language (UML) to create a use-case diagram for an e-commerce platform.
11. List the various types of software process models
12. Develop a formal specification for a simple calculator application.
13. Examine the differences between structural and behavioural models and assess their roles in system development

Unit 2

1. Identify the key practices in Extreme Programming (XP).
2. Analyze the benefits and drawbacks of using SCRUM over other Agile methodologies like ASD or DSDM in large teams.
3. Evaluate the effectiveness of iterative development cycles in SCRUM and how they impact project timelines.
4. Demonstrate how feature-driven development (FDD) can be applied to a product with a large number of small, independent features.

5. Use Adaptive Software Development (ASD) to plan and execute a project where requirements are highly uncertain.
6. Explain how Crystal methods can be applied to manage communication and collaboration in a small software development team.
7. State the phases in the SCRUM process
8. Create a feature list for a system using the Feature-Driven Development (FDD) approach.
9. Analyze how the principles of Agile development can be applied to industries other than software development, such as manufacturing or healthcare.
10. Describe the process flow in the SCRUM methodology.
11. Construct a plan for an Agile project that incorporates test-driven development (TDD) as part of the Extreme Programming methodology.
12. Develop a user story for a simple e-commerce platform and apply Agile techniques to estimate effort and prioritize features.

Unit 3

1. Summarize the steps involved in risk monitoring and mitigation.
2. Project management spectrum; project metrics; project planning- estimation and scheduling- PERT, CPM,
3. GERT, resource loading and resource levelling, types of project contracts from project management, agile
4. planning, risk mitigation and monitoring,
5. Construct a project control plan using earned value management (EVM) for tracking progress in a large-scale system development project.
6. Compare the advantages and disadvantages of resource loading versus resource leveling in the context of software project management.
7. Classify between CPM (Critical Path Method) and PERT in project management.
8. Analyze how project metrics such as cost variance, schedule variance, and earned value can be used to monitor progress and forecast project success.
9. Examine how the project management spectrum (traditional vs. Agile) influences the choice of scheduling techniques such as PERT and CPM.
10. Discuss the process of code cloning detection and its impact on software quality.
11. Analyze the relationship between quality management and project control techniques in ensuring the success of software projects.
12. Apply code refactoring techniques to eliminate code cloning issues in a legacy software system.

Unit 4

1. Identify the main stages in the CMMI (Capability Maturity Model Integration) process model.
2. Apply the principles of service-oriented architecture (SOA) to design a healthcare management system.
3. Apply mobile agents in an e-commerce platform. How would they handle customer requests while minimizing server load?
4. Demonstrate how mobile agents can be used in a distributed system to perform routine database maintenance tasks autonomously.
5. Use agents to develop a fault-tolerant system for a cloud-based file storage application. How would the agents manage failures in the system?
6. Apply a multi-agent system to handle dynamic scheduling of resources in a large-scale distributed computing environment.
7. Compare the differences between RESTful and SOAP-based web services and analyze their suitability for real-time applications.
8. Describe the characteristics that distinguish real-time software from other types of software.
9. Analyze the challenges of integrating real-time software systems into non-real-time environments, and propose solutions for managing such integration.
10. Compare the performance of monolithic and service-oriented architectures in handling high user loads.
11. Describe how mobile agents differ from traditional agents.
12. Apply the CMMI guidelines to pilot a new requirement management process in an organization. How will you evaluate its effectiveness?
13. Analyze how various software process improvement models (e.g., CMMI, ISO 9001) can be incorporated into an existing software development lifecycle to enhance product quality.