Chapter 1: Introduction to Agile

1. What are the key limitations of the Waterfall model that led to the emergence of Agile methodologies? How did iterative models evolve to support the limitations of traditional models?
2. What role does flexibility and customer collaboration play in Agile development? How does Agile address the rigidity of earlier models like Waterfall?
3. Why is rapid delivery considered a crucial aspect in modern software development?
4. What are the four core values of the Agile Manifesto and how do they differ from traditional software values?
5. Explain how responding to change is valued over following a plan in Agile. How do Agile principles promote customer satisfaction?
6. Provide examples of cultural resistance that teams may face when implementing Agile.
7. How does Agile promote team ownership and accountability?

Chapter 2: Agile Frameworks

1. What are the key responsibilities of the Scrum Master and how do they differ from a traditional project manager?
2. Compare Sprint Review and Sprint Retrospective in terms of purpose and output. How does Sprint Planning help in aligning the team's focus and goals?
3. What is backlog refinement and why is it important in Scrum?
4. Define Kanban and its core principles. Why is limiting Work In Progress (WIP) crucial in Kanban?
5. Explain how cycle time and lead time are measured and monitored in a Kanban system.
6. Describe the Red-Green-Refactor cycle in Test Driven Development (TDD). What are the roles in XP and how do they contribute to the development process?
7. List and explain the 7 Lean principles in software development.
8. Identify three types of waste in software projects and how to eliminate them. What is Kaizen, and how is it applied in Lean Software Development?

Chapter 3: Agile Project Management

1. What is a user story? Write an example and explain about the INVEST criteria. 2. Explain the difference between Acceptance Criteria and User Story. Describe how story points and the Fibonacci sequence help in estimation.
3. Define Minimum Viable Product (MVP) and explain its relevance in Agile. Describe roadmap planning and its role in long-term Agile project execution.
4. Explain how Burndown and Burnup charts are used in Agile project tracking. When should you use Burnup charts instead of Burndown charts?
5. Compare risk management approaches in traditional vs Agile projects. 6. Explain how Agile events like Daily Standups and Retrospectives help in identifying risks.
7. What is the impact/probability matrix and how is it used in Agile?

Chapter 4: Agile Tools and Technologies

1. What is the role of project management tools in Agile development? Describe key features to look for in Agile PM tools.
2. What are best practices for effectively using PM tools in Agile?
3. Differentiate between synchronous and asynchronous communication with examples. When should Agile teams choose chat, call, or documentation?
4. Define CI and CD and explain their importance in Agile.
5. What are the benefits of implementing CI/CD in Agile?
6. Describe the sequence of steps in a typical CI/CD pipeline.
7. Explain how version control integrates with CI tools.

Chapter 5: Agile Practices and Techniques

1. Define TDD and describe its cycle. What are the benefits and challenges of TDD?
2. Define BDD and explain how it evolved from TDD.
3. Compare and contrast BDD with TDD.
4. What is the Given-When-Then structure in BDD? How does BDD bridge the gap between technical and business teams?
5. Define Pair Programming and its core roles. Compare Expert-Novice and Ping-Pong pairing modes.
6. When should teams prefer Pair Programming over Mob Programming? Describe how remote pair programming works.
7. Define refactoring and its importance in Agile projects. List and explain common refactoring techniques.
8. What are the two types of code reviews and how do they differ? What mindset should reviewers have during a code review?