Agile Project Management (20AI&DS7403H)

Part-A (Compulsory Questions)

1. What are the core components of a good plan in Agile? **(CO1 K1 - Knowledge)**
2. How does Agile planning differ from traditional project planning? **(CO1 K1 - Knowledge)**
3. What are the benefits of using story points? **(CO2 K1 - Knowledge)**
4. Define ideal time in Agile. **(CO2 K1 - Knowledge)**
5. Explain what re-estimation is and when it is required. **(CO3 K1 - Knowledge)**
6. What are multiple levels of planning? **(CO2 K1 - Knowledge)**
7. How does Agile handle partially completed stories? **(CO3 K2 - Comprehension)**
8. Define Planning Poker as a technique in Agile. **(CO3 K1 - Knowledge)**
9. Why do estimates need to be shared within a team? **(CO3 K1 - Knowledge)**
10. How does multitasking lead to delays in project delivery? **(CO1 K1 - Knowledge)**

Part-B (Detailed Questions)

11. Explain the Agile Manifesto and its 12 principles. **(CO1 K2 - Comprehension)**
12. Compare Agile methodologies with traditional software development models. **(CO1 K4 - Analysis)**
13. Describe different Agile estimation techniques and their significance. **(CO2 K3 - Application)**
14. What is velocity in Agile, and how is it measured? **(CO2 K3 - Application)**
15. Explain the concept of a sprint backlog and its role in Agile development. **(CO3 K2 - Comprehension)**
16. Discuss the importance of continuous integration and deployment in Agile. **(CO3 K4 - Analysis)**
17. Compare Scrum, Kanban, and SAFe methodologies in Agile project management. **(CO4 K4 - Analysis)**
18. How does Agile ensure stakeholder collaboration and feedback in project execution? **(CO4 K5 - Evaluation)**
19. Explain the importance of risk management in Agile projects. **(CO5 K5 - Evaluation)**
20. How does Agile ensure continuous improvement and quality assurance? **(CO5 K6 - Creation)**

Software Engineering (20AI&DS6303 / 20CS6303)

Part-A (Compulsory Questions)

1. Define Software? **(CO1 K1 - Knowledge)**
2. List different domain applications in software engineering. **(CO1 K1 - Knowledge)**
3. Write the disadvantage of the waterfall model. **(CO1 K2 - Comprehension)**
4. What is traceability? **(CO2 K1 - Knowledge)**
5. Define use case? **(CO2 K1 - Knowledge)**
6. Define architectural pattern. **(CO3 K1 - Knowledge)**
7. Define modularity. **(CO3 K1 - Knowledge)**
8. Explain abstraction. **(CO3 K2 - Comprehension)**
9. Distinguish between testing and debugging. **(CO4 K3 - Application)**
10. Give some flow graph notations. **(CO4 K3 - Application)**

Part-B (Detailed Questions)

11. Explain the software development life cycle (SDLC) and its phases. **(CO1 K2 - Comprehension)**
12. Compare different software process models with their advantages and disadvantages. **(CO1 K4 - Analysis)**
13. Explain functional and non-functional requirements with examples. **(CO2 K3 - Application)**
14. What is requirement engineering? Explain its different phases. **(CO2 K3 - Application)**
15. Describe different software architectural styles with examples. **(CO3 K4 - Analysis)**
16. What are design patterns? Explain different types of design patterns in software engineering. **(CO3 K4 - Analysis)**
17. Explain the different levels of testing in software engineering. **(CO4 K3 - Application)**
18. Discuss software quality assurance (SQA) and its importance. **(CO4 K5 - Evaluation)**
19. What are different types of software maintenance? Explain with examples. **(CO5 K4 - Analysis)**
20. Explain the importance of risk management in software projects. **(CO5 K5 - Evaluation)**