Question 3:

Testing in an Agile Framework - Outline:

1. General Testing Concepts in Agile:

Continuous Collaboration: Testing is integrated throughout the development cycle with constant communication between QA, developers, and product owners.

Test-Driven Development (TDD): QA often works closely with developers to ensure that tests (unit, integration) are written before code is implemented.

Behaviour-Driven Development (BDD): Encourages collaboration with stakeholders, where test cases are written in natural language and focus on business behaviour.

2. QA Team Process in Agile:

a. Sprint Planning: QA attends sprint planning to understand requirements, user stories, and acceptance criteria.

b. Test Design: Test cases are derived from user stories, focusing on both functional and non-functional aspects. Exploratory testing also plays a key role.

c. Test Execution: Testing is done iteratively, with manual and automated tests being executed as development progresses.

d. Automation: QA works to automate repetitive regression and smoke tests to improve efficiency.

e. Review and Retrospective: QA provides feedback during sprint reviews and retrospectives to improve processes.

3. When to Write Test Cases & Who Should See Them:

When:

Test cases should be written early during the sprint (typically after user stories are created and acceptance criteria are defined).

QA participates in backlog grooming to create preliminary tests even before development starts.

Who Sees Them?

Test cases should be shared with developers, product owners, and other relevant stakeholders to ensure alignment on testing goals and expectations.

Test cases are stored in a shared location for visibility.

4. Regression Testing:

Continuous Regression: Regression tests are executed on each build to ensure that new code changes do not negatively impact existing functionality.

Automation: Regression tests should be automated for faster feedback and to ensure consistency across builds.

Test Coverage: Select and prioritize the most critical and frequently used areas of the application for regression.

5. Executing Test Cases & Reporting Results

Execution:

Test cases should be executed in small increments (e.g., daily or after each feature development) for faster feedback.

Both manual and automated tests are executed based on the test case types.

Reporting:

Results are logged in a test management tool (e.g., Jira, TestRail) with detailed information about the pass/fail status, defects found, and severity.

Quick and concise reports are generated for daily stand-ups and sprint reviews.

6. Prioritizing Test Cases:

Risk-Based Prioritization: Focus on critical areas that have the highest impact on users or business operations.

Frequency of Use: Prioritize test cases based on how frequently a feature is used by customers.

New Features and Changes: Give priority to testing new features, changes, or bug fixes that are being developed in the current sprint.

Regression Tests: High-priority tests should be automated and run frequently during regression.

7. Deciding What Test Cases to Write:

a. Acceptance Criteria: Test cases should cover all acceptance criteria outlined for user stories.

b. Edge Cases: Consider boundary conditions and error handling scenarios.

c. Exploratory Testing: Include test cases for exploratory tests to uncover unforeseen issues.

d. Non-Functional Testing: Ensure coverage of performance, security, and usability aspects.

e. Stakeholder Input: Gather input from product owners and end-users to focus on critical features and user journeys.

8. What should be Included in Test Cases

a. Test Case ID: Unique identifier for each test case.

b. Test Description: Clear description of the test scenario.

c. Pre-conditions: Any setup needed before running the test.

d. Test Data: Input data needed for the test.

e. Steps to Execute: A clear step-by-step guide on how to perform the test.

f. Expected Result: What the system should do if the test passes.

G .Actual Result: What actually happened during testing (to be filled in after execution).

H .Pass/Fail Status: Outcome of the test.

i. Defects/Issues: If any, provide details of the defect or issue found during the test.