Geric Gultiano 3/10/2025
BSIT-3A

TLA #4: QA Process and Testing Approach

The Deming Cycle (also known as the Plan-Do-Check-Act (PDCA) Cycle) is a continuous improvement process widely used in Quality Assurance (QA) and software development. It consists of four distinct phases:

1. **Plan**:
    a. In this phase, you define what needs to be achieved and how to achieve it. You identify project requirements, set goals, and plan testing strategies. For software QA, this involves defining the testing scope, preparing test cases, and determining test objectives and criteria for success.

    b. Example:
        i. In the planning phase of a software project, a team identifies the functionalities to test (e.g., login page, user registration) and decides on testing techniques (manual or automated) and test environments.

2. **Do**:
    a. The "Do" phase is where the testing activities are executed. Test cases are run, and the system is tested according to the plan. This phase involves implementing the tests, recording results, and documenting any issues or defects that are discovered during testing.

    b. Example:
        i. During the "Do" phase, the QA team executes the predefined test cases, such as validating login functionality, checking data entry forms, and ensuring that all navigation works correctly.

3. **Check**:
    a. After executing the tests, the next step is to check and analyze the results. This involves comparing the actual test results with the expected results, identifying defects, and assessing the effectiveness of the testing process. In this phase, QA teams evaluate if the project is meeting the quality standards set in the "Plan" phase.

    b. Example:

        i. If a test case fails (e.g., the login page doesn’t load), the QA team documents the issue, compares the failure to the expected outcome, and classifies it as a defect that needs to be addressed.

4. **Act**:
    a. In the "Act" phase, actions are taken based on the findings from the "Check" phase. The team may decide to fix defects, improve processes, and refine testing strategies for future iterations. This phase is about continuous improvement to enhance the overall quality and prevent recurring issues.

    b. Example:
        i. After finding that certain bugs consistently affect the login page, the development team may refine the code or improve the testing process to ensure these issues don’t occur again.

Software Testing Approach

A software testing approach refers to the strategy used to validate that a software product meets its requirements and functions as intended. Here's an example of a testing approach used in software QA:

Agile Testing Approach:

1. **Test-Driven Development (TDD)**: In an Agile environment, developers and testers work collaboratively in short, iterative cycles. TDD focuses on writing test cases before coding, ensuring that each feature is tested as it is developed. Unit tests are written first, followed by the code that passes these tests. This ensures immediate feedback and helps prevent defects from entering the system.

2. **Continuous Integration (CI) and Continuous Testing**: Automated tests are executed as part of a continuous integration pipeline. Every change made to the codebase triggers automated tests to verify that the new code doesn't break existing functionality. This ensures that testing occurs frequently and early in the development process.

3. **Exploratory Testing**: In addition to scripted test cases, testers engage in exploratory testing where they actively explore the software, using their intuition and experience to find defects that might not be captured by predefined test scenarios. This is particularly useful for uncovering edge cases or issues that weren't foreseen during test planning.