Software Test Plan

Software QA & Testing, SE, BZU, Dr. Samer Zein

Introduction

A Test Plan in software testing is a comprehensive document that outlines the strategy, scope, objectives, resources, and schedule for testing a particular software application. It serves as a roadmap for the testing process and is essential for efficient and effective testing.

Components of Software Test Plan:

Introduction:

- Objective: Begin by stating the purpose of the test plan.
  - Example: The objective of this test plan is to ensure the quality and reliability of our e-commerce web application, "WebShop."

Scope:

- Inclusions: Define what aspects of the web application will be tested.
  - Example: This test plan covers functional, performance, security, and usability testing.
- Exclusions: Specify what won't be tested.
  - Example: Mobile applications and third-party integrations are excluded from this test plan.

Components of Software Test Plan:

Test Objectives:

- **Functional Testing Objectives:** List specific functional testing objectives.
    - Example: Verify that users can create accounts, add items to the shopping cart, and complete the checkout process without errors.
- **Performance Testing Objectives:** State the performance-related goals.
    - Example: Ensure that the web application can handle 1,000 concurrent users with a response time of under 2 seconds.
- **Security Testing Objectives:** Specify security goals.
    - Example: Identify and mitigate vulnerabilities such as SQL injection and cross-site scripting (XSS) attacks.
- **Usability Testing Objectives:** Describe usability testing goals.
    - Example: Evaluate the user interface for ease of use and ensure compliance with accessibility standards.

Components of Software Test Plan:

Test Strategy:

- **Describe the overall approach to testing, including methodologies, testing levels (unit, integration, system, etc.), and techniques.**
    - **Example: We will follow an Agile testing approach with a combination of manual and automated testing. Unit testing will be performed by the development team, while system testing and user acceptance testing will be conducted by the QA team.**

Test Schedule:

- Provide a timeline for testing activities, including start and end dates for each testing phase.
    - Example: Functional testing will begin on [Start Date] and conclude on [End Date]. Performance testing is scheduled for [Start Date] to [End Date].

Components of Software Test Plan:

Test Resources:

- List the resources required for testing, including hardware, software, personnel, and tools.
  - Example: Test environments include a staging server and a separate database server. Testing tools to be used: Selenium for automated testing, JMeter for performance testing. The testing team comprises [List of Team Members].

Test Deliverables:

- Specify the documents and reports that will be generated during testing.
  - Example: Test cases, test data, defect reports, test summary reports, and a final test report will be produced as deliverables.

Components of Software Test Plan:

- **Risks and Assumptions:**
  i. Identify potential risks to the testing process and the assumptions made.
     1. Example: Assumption: The staging environment will mirror the production environment. Risk: Limited availability of test data for specific scenarios.
- **Dependencies:**
  i. List any external dependencies that may impact the testing process.
     1. Example: The testing schedule is dependent on the completion of development tasks in the sprint.
- **Exit Criteria:**
  i. Define the conditions that must be met for testing to be considered complete.
  ii. Example: All high-priority defects must be resolved, and the web application must pass a final regression test with no critical issues.
- **Approvals:**
  i. Specify who needs to approve the test plan.
  ii. Example: The test plan requires approval from the project manager and the QA manager.
- Appendices: Include any supplementary information, such as glossaries, test data, or references.

Test Plan in Agile Software Development (SCRUM)

- Test plan formation in Scrum is typically done at the beginning of each sprint, during the sprint planning meeting.
- This allows the Scrum team to identify and prioritize the test cases that need to be executed in order to verify the functionality of the user stories that are planned for the sprint.
- However, it is important to note that the test plan is not a static document. It should be updated and refined throughout the sprint as needed.
- For example, if the team discovers new bugs or requirements, they may need to add additional test cases to the plan

The Process of Test Plan Formation in SCRUM

Before Sprint Planning

The Scrum Master and the testers should meet to review the Product Backlog and identify the user stories that are planned for the sprint. They should also identify any risks or challenges that may need to be addressed during testing.

During Sprint Planning

The Scrum team should review the test plan and make any necessary updates. They should also prioritize the test cases and assign them to team members.

Throughout the Sprint

The Scrum team should execute the test cases and update the test plan as needed.

At the end of Sprint

The Scrum team should review the test results and identify any remaining bugs. They should also update the test plan for the next sprint.

Best Practices for TP Formation in SCRUM

- **Involve the entire Scrum team:** The test plan should be developed and reviewed by the entire Scrum team, including the developers, testers, and Product Owner. This will help to ensure that the plan is comprehensive and that everyone is on the same page.
- **Keep the test plan concise and focused:** The test plan should be concise and focused on the specific test cases that need to be executed in order to verify the functionality of the user stories that are planned for the sprint.
- **Update the test plan regularly:** The test plan should be updated and refined throughout the sprint as needed. This will help to ensure that the plan is accurate and that the team is testing the latest version of the software.
- **Use a test management tool:** A test management tool can help the Scrum team to organize and track their test cases, as well as to generate reports.

Test Plan in Traditional SW Development Projects

1. **The test plan is created by the test team, in collaboration with the developers and other stakeholders.** The test plan should identify the scope of testing, the test approach, the test cases, the test data, the test environment, and the test schedule.
2. **The test plan is reviewed and approved by the project manager and other stakeholders.** This ensures that the test plan is comprehensive and that it meets the needs of the project.
3. **The test team executes the test cases and reports the results.** The test results are used to identify any defects in the software.
4. **The defects are fixed by the developers and the test team re-tests the affected areas.** This process continues until all of the defects have been fixed and the software is ready for release.