Testing Lifecycle



Prepared by

**Fasttrack**

Solution Architect

Contributors

**List of Contributors**

Revision and Signoff Sheet

Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Change Reference |
|  |  | 1.0 | Initial draft for review/discussion |
|  |  |  |  |
|  |  |  |  |

Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version Approved | Position | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

Table of Contents

[1 Introduction 5](#_Toc128734610)

[1.1 Need for Testing 5](#_Toc128734611)

[1.2 Planning 6](#_Toc128734612)

[1.3 Design 7](#_Toc128734613)

[1.4 Environment Setup 8](#_Toc128734614)

[1.5 Execution 10](#_Toc128734615)

[1.6 Reporting 11](#_Toc128734616)

1. Introduction

In this document we will provide an overview of the Testing Lifecycle process.

Testing Lifecycle refers to the process of planning, designing, executing, and reporting on software testing activities throughout the software development lifecycle. It involves a series of activities aimed at ensuring the software meets its requirements and quality standards. The testing lifecycle typically includes phases such as test planning, test design, test environment setup, test execution, test reporting, and test closure. By following a proper testing lifecycle, organizations can improve the quality of their products, reduce the overall cost of testing, and increase customer satisfaction.

* 1. Need for Testing

Software testing is an important part of the software development lifecycle because it ensures that the software being developed meets the requirements and works properly.

Here are some of the main reasons why software testing is required:

1. **Identify defects**: Testing assists in identifying defects and errors in software, which can lead to potential system failures or problems. These issues can be addressed and resolved before the software is released to end users if they are identified early in the development cycle.
2. **Improve quality**: Testing helps to improve software quality by ensuring that it meets the requirements, functions as expected, and performs optimally under various conditions.
3. **Ensure reliability**: Testing ensures that the software is dependable and consistently performs over time. This is especially true for software used in mission-critical systems or applications.
4. **Increase customer satisfaction**: Testing ensures that the software meets the needs and expectations of end users, which can lead to higher levels of customer satisfaction.
5. **Lower costs**: Testing helps identify defects and errors early in the development cycle, which can lower the costs of fixing these issues later in the development process or after release.
6. **Maintain reputation**: Testing contributes to the software's and development organization's reputation by ensuring that the software meets end-user quality standards and expectations. This can aid in the development of trust and loyalty among customers and stakeholders.
   1. Planning

The Planning phase is the first phase of the testing lifecycle and involves defining the testing process's objectives, scope, approach, and deliverables. During this phase, the testing team collaborates closely with the development team and stakeholders to identify testing requirements and set the testing effort's direction.

The following are the primary objectives of the Test Planning phase:

1. **Define the testing objectives**: This entails determining the purpose and goals of the testing procedure. It entails determining the types of testing that will be performed to ensure that the software meets its requirements and quality standards, such as functional testing, performance testing, and security testing.
2. **Identify the scope of testing**: Define the features, components, and systems that will be tested as part of the testing effort. It also includes determining the testing process's timeline as well as any constraints or risks associated with the testing effort.
3. **Define the testing approach**: This involves determining the approach that will be used for the testing process, including the testing techniques, tools, and methods that will be used to carry out the testing.
4. **Develop the testing plan**: This involves creating a comprehensive testing plan that outlines the testing approach, test schedule, resources, and deliverables. The testing plan serves as a roadmap for the entire testing process.
5. **Establish communication and reporting channels**: This involves defining the communication and reporting channels between the testing team, the development team, and stakeholders. It ensures that everyone is aware of the progress of the testing process and any issues that arise are addressed promptly.

Planning establishes the direction for the entire testing process and ensures that the testing effort is aligned with the software development project's business goals and objectives. It serves as the foundation for the testing process and allows the testing team to carry out testing activities in an effective and efficient manner.

* 1. Design

The next phase of the testing lifecycle is the Test Design phase, which focuses on designing the test cases and scenarios that will be used to verify the software under test.

The following are the primary objectives of the Test Design phase:

1. **Analyze the requirements**: In this phase, the testing team analyzes the software requirements and identifies the functional and non-functional requirements that need to be tested. The testing team also reviews the requirements for clarity and completeness and identifies any ambiguities or gaps that may impact the testing effort.
2. **Identify test scenarios**: Based on the requirements, the testing team identifies the test scenarios that need to be tested. Test scenarios are a sequence of steps that a user would take to perform a specific task in the software. The testing team also identifies the expected results for each test scenario.
3. **Create test cases**: In this phase, the testing team creates the test cases that will be used to verify the software. Test cases are specific conditions or situations that need to be tested to validate the software functionality. Test cases are typically documented in a test case specification document, which includes the test case description, preconditions, test data, steps, expected results, and actual results.
4. **Design test data**: The testing team designs the test data that will be used to execute the test cases. Test data is the input data that is used to test the software. The testing team ensures that the test data covers all possible scenarios and is representative of the real-world data.
5. **Define test automation strategy**: The testing team defines the test automation strategy, which includes identifying the test cases that can be automated and selecting the appropriate tools and frameworks to automate the testing process. The testing team also develops the test automation scripts that will be used to execute the automated tests.

The design phase ensures that the testing team has a comprehensive set of test cases and scenarios that cover all the software's requirements. It also ensures that the test data is representative of real-world data and that the testing process can be automated as needed. Organizations can ensure that the software is thoroughly tested and that all defects or issues are identified and addressed before the software is released to the market by following a structured Test Design phase.

* 1. Environment Setup

The third phase of the testing lifecycle is Test Environment Setup, which focuses on configuring the testing environment for the software under test. The hardware, software, and network infrastructure required to run the tests are all part of the testing environment.

The following are the primary objectives of the Environment Setup phase:

1. **Identify testing infrastructure requirements**: In this phase, the testing team identifies the hardware and software infrastructure required to execute the tests. This includes identifying the servers, workstations, operating systems, databases, and other software components required to support the testing process.
2. **Configure testing infrastructure**: The testing team sets up and configures the testing infrastructure. This includes installing and configuring the hardware and software components required for testing.
3. **Install and configure the software under test:** The testing team installs and configures the software under test in the testing environment. The testing team ensures that the software is installed correctly and that it is configured to work with the testing infrastructure.
4. **Define test data:** The testing team defines the test data that will be used to execute the tests. This includes identifying the test data sources and ensuring that the data is available in the testing environment.
5. **Establish monitoring and reporting mechanisms**: The testing team establishes the monitoring and reporting mechanisms required to track the progress of the testing process. This includes setting up monitoring tools to track the performance of the testing infrastructure and establishing reporting mechanisms to report on the test results.

This phase ensures that the testing environment is set up correctly and that the software under test is installed and configured to work with the testing infrastructure. It also ensures that the test data is available in the testing environment and that the monitoring and reporting mechanisms are established to track the progress of the testing process. By following a structured Test Environment Setup phase, organizations can ensure that the testing process is executed efficiently and effectively, and that the software is thoroughly tested in a controlled environment.

* 1. Execution

Test Execution is the fourth phase of the testing lifecycle and is focused on executing the test cases and scenarios that were designed in the Test Design phase.

The following are the primary objectives of the Execution phase:

1. **Execute test cases**: In this phase, the testing team executes the test cases that were designed in the Test Design phase. The testing team follows the test case steps and records the results in a test execution report.
2. **Log defects**: During the test execution, the testing team logs any defects or issues that are identified in the software. The testing team records the details of the defect, including the steps to reproduce the issue, the expected and actual results, and the severity of the defect.
3. **Retest defects**: Once a defect is logged, the testing team works with the development team to fix the defect. After the defect is fixed, the testing team retests the defect to ensure that it has been resolved.
4. **Update test case results**: The testing team updates the test case results in the test execution report based on the results of the test execution.
5. **Monitor progress**: The testing team monitors the progress of the testing process and tracks the test execution metrics, such as the number of test cases executed, the number of defects logged, and the defect resolution rate.

This phase ensures that the testing team executes the test cases and scenarios that were designed in the Test Design phase, and that any defects or issues in the software are identified. It also ensures that the testing team collaborates with the development team to resolve any defects and that the defects are retested to ensure that they have been resolved. Organizations can ensure that the software is thoroughly tested and that all defects or issues are identified and addressed before the software is released to the market by following a structured Test Execution phase.

* 1. Reporting

The final phase of the testing lifecycle is reporting, which focuses on creating reports that summarize the testing activities and results. This ensures that all testing activities and results are properly documented and summarized in an understandable and concise manner. It also ensures that the software development process is improved based on the results and analysis of the tests. Organizations can ensure that they have a clear understanding of the testing activities and results by following a structured Reporting Phase, allowing them to make informed decisions about the software's release.