

Quality Assurance Team

TABLE OF CONTENTS



Start adding headers to the doc to get a table of contents.

Introduction

QA (Quality Assurance) is an important process in software development that ensures the quality of a software product. In a software company, QA is responsible for designing, implementing, and maintaining a quality management system that ensures the software meets the requirements and standards set by the customer or the company.

The role of QA in a software company includes:

1. Testing: The primary responsibility of QA is to test the software to identify bugs, errors, and other issues that may impact the functionality or user experience.
2. Documentation: QA is also responsible for documenting the testing process, test results, and any bugs or issues identified during the testing process.
3. Process improvement: QA is responsible for continuously improving the software development process, identifying areas of improvement, and implementing changes to improve the overall quality of the software.
4. Standards compliance: QA ensures that the software is developed in compliance with industry standards and best practices.
5. Customer satisfaction: QA also plays a critical role in ensuring customer satisfaction by identifying and addressing any issues or concerns raised by customers.



<https://youtu.be/QJqNYhiHysM>

Steps In QA :

Quality Assurance (QA) is the process of ensuring that software products are developed and delivered with high quality. A well-designed QA process can help software companies improve

their product quality, increase customer satisfaction, and reduce costs associated with rework and defect resolution.

Here are the steps involved in a typical QA process in a software company:

1. **Planning:** The first step in the QA process is to develop a plan that outlines the QA objectives, the scope of testing, the resources needed, and the timeline for completing the testing.
2. **Requirements gathering:** The next step is to gather and review the requirements for the software product. This helps to ensure that the product meets the customer's needs and expectations.
3. **Design and review:** Once the requirements have been gathered, the design of the software is created, and reviewed for accuracy and completeness.
4. **Test case development:** Test cases are developed based on the software requirements and design. These test cases define the expected behavior of the software under various conditions.
5. **Testing:** The testing phase involves executing the test cases and verifying that the software behaves as expected. This includes functional testing, performance testing, and other forms of testing as necessary.
6. **Defect tracking:** Any defects or issues found during testing are documented and tracked in a defect tracking system. This helps to ensure that all issues are addressed and resolved.
7. **Regression testing:** After defects are resolved, regression testing is performed to ensure that no new issues have been introduced.
8. **Release:** Once testing is complete, and the software is deemed to be of high quality, it is released to customers.
9. **Continuous improvement:** Finally, the QA process is continuously evaluated and improved to ensure that the software company's products continue to meet high-quality standards.

Activities In QA

1. **Test Planning:** The QA team creates a comprehensive test plan that outlines the scope of testing, testing methodologies, and test cases to be executed.
2. **Test Case Design:** The QA team designs test cases based on the software requirements and specifications to ensure that all features and functionalities are tested.
3. **Test Execution:** The QA team executes test cases and scripts to identify defects, bugs, and other issues in the software.
4. **Defect Reporting and Management:** The QA team reports defects, bugs, and issues identified during testing and manages them through their resolution.

5. Test Automation: The QA team develops and maintains test automation frameworks and scripts to improve testing efficiency and accuracy.
6. Continuous Improvement: The QA team continuously monitors and improves the software development process to ensure that it meets the highest quality standards.
7. Documentation: The QA team documents the testing process, test results, and any bugs or issues identified during the testing process.
8. Release Management: The QA team plays a critical role in ensuring that the software release process is smooth, error-free, and meets the requirements and expectations of customers.

Software Testing Life Cycle :

Software testing is at the core of exceptional digital experiences. If you can't sufficiently and thoroughly validate your products before customers use them, you risk alienating them. Software testing isn't just a stage before release; it must be a multi-faceted, organization-wide, ongoing effort.

Just as there is a software development life cycle (SDLC) for developing digital products, there is a software testing life cycle (STLC) for validating them. Different members of the organization get involved at different stages to help businesses achieve their **quality goals** in a strategic, documented manner.

Software Testing Life Cycle Phases :

1.Requirement Analysis :

Most development initiatives begin with software requirements that specify what the business expects from the project. Software requirements often include high-level business needs, architectural requirements that detail how the feature will be designed and supported, and detailed system requirements from which developers build the product. System requirements include functional and non-functional specifications, both of which present opportunities to test and validate.

In this STLC phase, testers work both within their own teams and cross-functionally to contextualize how they will test the software. Requirement analysis often includes brainstorming sessions, identifying blind spots or unclear areas in the requirements, and prioritizing certain assessments.

When in doubt or lacking requirements documentation, the QA team will question the engineering or business side to clarify and calcify a testing strategy

1. Review Requirements Documentation: The QA team should review the requirements documentation, including functional requirements, non-functional requirements, and design specifications to gain a clear understanding of the software's scope and purpose.
2. Analyze Requirements: The QA team should analyze the requirements documentation to identify potential gaps, ambiguities, and conflicts that may impact the testing process.
3. Communicate with Stakeholders: The QA team should communicate with stakeholders, including the project manager, developers, and business analysts, to clarify any questions or concerns regarding the software requirements.

Please find Below Sample Documents from Business Analyst :



Business Requirement Document.docx

15KB, Uploaded a week ago



Detail field document.docx

14KB, Uploaded a week ago



Functional Specification Document.docx

14KB, Uploaded a week ago



Functional Requirement Document.docx

14KB, Uploaded a week ago



In Scope & Out of Scope.docx

14KB, Uploaded a week ago



Non Fuctional Requirement Document.docx

15KB, Uploaded a week ago





Process flow.docx

14KB, Uploaded a week ago



2. Test planning

The second STLC phase is important, as it guides much of the work to follow. Test planning takes the insights found during requirements or product analysis and turns them into a documented QA strategy.

The test team leadership determines what resources and efforts will evaluate the release. The resulting test plan documentation both informs testers and other departments how the testing work will commence, keeping everyone on the same page. This plan is especially helpful if other members of the organization will take part in testing and bug remediation, such as developers executing unit tests and writing hotfixes.

The test plan spells out several details of the QA work to be done, including the scope, objectives, types of functional and non-functional tests (both automated and manual), and details for the test environments. Once these details are determined, **test management** sets roles and timelines for the work. Finally, the testing team can determine what deliverables it will provide upon completion of the STLC phases.



Test Plan.docx

18KB, Uploaded a week ago



3. Test case design and development

With the test plan in place, testers can begin to write and create detailed test cases. In this STLC phase, the QA team fleshes out the details of the structured tests they will run, including any test data they will need to facilitate those tests. While tests must ultimately validate the areas defined by requirements, testers can exert their skills and creativity in *how* they achieve this task.

When conceptualizing test cases, the tester's goal should be to validate functionality within the allotted time and scope, especially core functionality. Test cases should be simple and well understood for any member of the team, but also unique from other test cases. Test cases should aim to achieve full coverage of the requirements in the specifications document — a traceability matrix can help track coverage. It's important that test cases be identifiable and repeatable, as developers will add new functionality to the product over time, requiring tests to run again. They must also not alter the test environment for future tests, especially when validating configurations.

Test cases might also require maintenance or updates over time to validate both new and existing functionality. This work also occurs at this STLC stage.

Once test cases are ready, a test team lead or peer can review them. They might also review and update automated test scripts at this STLC stage. Ultimately, the team prioritizes and organizes these test cases into test suites that run



Test Case Specification.docx

14KB, Uploaded a week ago



test scenario.docx

13KB, Uploaded a week ago



4. Test environment setup

The test environment provides the setting where the actual testing occurs. This is a crucial software testing life cycle phase, and it requires help from other members of the organization. Testers must have access to bug reporting capabilities, as well as the application architecture to support the product. Without these elements, testers might not be able to do their jobs.

Once ready, testers establish the **parameters for the test environment**, which include the hardware, software, test data, frameworks, configurations and network. In this STLC phase, testers adjust these environment parameters depending on what the test case requires. For example, the majority of a product's users might be on an Android device, use a certain version of a Chrome browser and have a certain amount of processing power on those devices — these are parameters the test environment would include.

Smoke tests within these test environments provide a very early and rudimentary check that the software is ready for more comprehensive testing. These smoke tests against the builds are part of the deliverable in this STLC phase.

5. Test execution

Next in the software testing life cycle, it's time to fully test the product. At this STLC stage, testers execute all of the test cases, or as many as is possible within the allotted time. QA professionals and automated scripts execute a number of functional and non-functional tests. Here in the STLC, testers will identify and **report detailed bugs** that arise from test case execution and log the system's performance compared to its requirements. As developers make fixes, testers often retest the product to make sure new defects don't materialize. With all of these tests piling up in the test execution STLC phase, it's important to make use of test automation where possible to achieve the test coverage and velocity you need.



Requirement Traceability Matrix.xlsx

8KB, Uploaded a week ago



Test execution log.doc

39KB, Uploaded a week ago



Defect Report.xlsx

10KB, Uploaded a week ago



6. Test cycle closure

The final STLC phase is test cycle closure. In this stage, the testing team provides a test closure report, which summarizes and communicates its findings with the rest of the team. This report typically includes summaries of the testing work and results, an assessment of the testing and the manager's approval.

During the test cycle closure, the testing team checks its deliverables, which include details relevant to the testing work, such as the test strategy, test case documents, automated test scripts and test results. The team will then complete and close incident reports, which detail unusual or unexpected behavior that the test team observes during testing. The team must also archive the resources it used during testing, such as scripts, tools and environments, for later use.

From there, the organization plans the product **for support and release**, which often includes acceptance and feedback from customer representatives.

Communication is key in this STLC phase, as additional perspectives might uncover a quality, cost or coverage issue that the rest of the group missed. These discussions can yield additional analysis or inform how to improve QA work in the future.



Test Summary Report.docx

56KB, Uploaded a week ago

