

What does quality assurance mean in software testing?

Most times, quality assurance (QA), quality control (QC), and testing are used interchangeably. Although they are all closely related and required as part of quality management, they are each entirely different processes. While QA overlooks the entire software development cycle, QC is a software engineering process that is a distinct part of the software testing cycle. In this article, we explain the difference between these terms and dive deeper into Software QA.

What you will learn:

- What is quality assurance?

- What is Quality Control?

- What is the difference between QA, QC, and Testing?

- So what is quality assurance in software testing?

 - What are the steps involved in Software Testing QA?

- What are different types of software testing?

- Techniques in software testing

- Why is QA so important in software testing?

- What is software quality assurance (SQA)?

 - SQA Plan

 - SQA Activities

- When is the best time to start software QA testing?

- When to stop the Software QA process

- FAQ

What is quality assurance?

Defining quality assurance begins with an understanding of what quality and assurance mean. Quality refers to how well something meets expectations, requirements, and standards. In the case of the product, quality describes how well it meets the needs and expectations of customers, in terms of design, durability, functionality, price, and reliability. Assurance is simply a positive declaration that gives confidence. In the case of the product, assurance guarantees that it will work as required and expected, without any glitches.

With that said, quality assurance is a systemic process used to determine whether a product or service meets the needs and expectations of customers.

Quality can only be achieved with the setup of a quality control mechanism.

What is Quality Control?

Quality control is the process through which product quality is achieved or improved. This entire process is one that incorporates the activities employed to address waste in the software development lifecycle (SDLC). Quality control functions involve testing activities used to discover and resolve technical issues. In conjunction with the development processes, these are followed throughout the SDLC.

What is the difference between QA, QC, and Testing?

QA is process oriented as it aims to prevent defects by improving the process. QC on the other hand is end-product-focused with the aim of detecting and correcting defects. QA in no way eliminates the need for QC as they both lie at the heart of quality management. QA precedes QC in the phases of software quality management. Testing forms part of QC and QA ensures that testing is done the right way.

So what is quality assurance in software testing?

Software testing is a process used to verify that a software product meets specified requirements. It aims to improve the software development process to make it effective and efficient by identifying errors, bugs, and gaps that conflict with the requirements. Quality assurance is also generally referred to as quality assurance testing.

What are the steps involved in Software Testing QA?

We believe that the software testing life cycle is the best way to explain the QA process. There are six main stages involved in the software development life cycle:

1. Requirement analysis

This is the first stage in the software development life cycle. It is the process through which testable requirements are identified by testers. QA professionals interact with all stakeholders for an in-depth understanding of their requirements.

There are many times when requirements are unclear or missing, so without ideal requirements for the project, the software tester needs to utilize tools and methods to organize testing.

A feasibility study of automation is also performed at this point, if necessary.

Requirement analysis forms the basis for project plans and test plans.

2. Test planning

At this stage is where it becomes necessary to:

- Devise a testing strategy;
- Identify risks;
- Choose the most suitable testing tools;
- Allocate roles in the team.

All of these are recorded as part of the test strategy in the test plan.

3. Test case development

As soon as the test plan is fully developed, the QA team begins the development of test cases. This phase's principal objective is the preparation of test cases for an individual unit or feature of the project. These structural and functional test cases cover the areas mentioned in the test plan: points of verification, validation, and the product's functionality. These test cases help the team get acquainted with the program without having to read the entire code.

4. Testing environment setup

This is an essential step in the software testing life cycle. It makes all the preparations required for an effective testing process. This includes setting up a network, a test server, test devices and PCs, as well as the generation of test data.

The test team is not necessarily involved in this process, however, they should be prepared to conduct a readiness check to ensure that the configuration is testable before being released for further testing. They need to also ensure the entry of the necessary test data into the system so that it is ready to be used.

This phase is sometimes completed together with the test case development phase.

5. Test execution

When the environment is set up, the test strategy is established, and the test plan is approved, it is now time to run the tests. The majority of them will be exploratory testing guided by the test scenarios. Using this method allows the largest number of defects to be found in the shortest possible time.

When the tests are performed by the operators and users, they use the test cases. These testers execute each test and compare the results to the expected results, assigning a pass, fail, or skip rating. The fail rating requires the tester to document what really happened during the test. In this phase, bugs are also logged into the bug tracking system identified in the test plan. Once the development team fixes the defect, the test case can be rerun in accordance with your test plan.

6. Cycle closure

This is the final phase of the software testing life cycle where upon execution of all test cases, the QA lead confirms that all required testing has been completed. Here, an analysis is conducted of the test cases - how many passed, failed or were skipped, defects found, and other metrics. This creates an opportunity for the team to improve future testing projects.

What are different types of software testing?

The types of software testing generally fall into two groups:

1. Manual Testing

In this process, test cases are executed without the use of automation tools, and from the point of view of the end-user. A human sits in front of a computer carefully carrying out the test steps to ensure that the application works as expected, based on the requirement document.

The various stages of manual testing include [unit testing](#), [integration testing](#), [system testing](#), and [user acceptance testing](#).

Manual testing also includes exploratory testing, which is where testers explore the software to identify any errors in it.

2. Automation testing

Automation testing, which is also referred to as test automation, is where manual testing is automated to facilitate quicker constant re-runs of test scenarios. It is a software solution that utilizes special automated testing software tools.

With the exception of regression testing, automation testing is also utilized in performance, load, and stress testing. Compared with manual testing, automation testing saves time and money, improves accuracy, and increases test coverage. It must be noted, however, that automation does not aim to eliminate manual testing altogether, but rather to reduce the number of test cases to be run manually.

Types of automation testing include:

- [Functional Testing](#)
- [Unit Testing](#)
- [Integration Testing](#)
- [Smoke Testing](#)
- [Non-functional Testing](#)
- [Performance Testing](#)
- [Regression Testing](#)
- [Keyword-driven Testing](#)
- [Data-driven Testing](#)

Techniques in software testing

[Black Box Testing](#) and [White Box Testing](#) are the two major categories of software testing.

While Black Box Testing and White Box Testing have different functionalities, they both guarantee the best version of the software and that there are no glitches in the system.

Why is QA so important in software testing?

To achieve high-quality software, a robust QA system is imperative. To be truly effective, [best practices](#) must be incorporated into the mix. Also, the [PDCA technique](#) (Plan, Do, Check, Act), when closely followed improves the quality of mobile testing or web testing, with each subsequent project.

QA is a proactive and preventative process with the following benefits:

1. It saves time and helps reduce cost

With a QA system in place throughout your production lifecycle, you are better able to predict and prevent issues like bugs, errors, and defects. The identification of problems in the initial stages enables you to arrange fixes early so that you save the time and money that would have been spent fixing them in later stages. Additionally, any errors or bugs that may have slipped through the QA process would probably be relatively minor and easy to fix.

2. It helps maintain product quality

QA, as a proactive process, facilitates the maintenance of product stability and reliability. Because it predicts issues, the Quality Assurance team, together with the Quality Control team is able to run tests at various stages of the lifecycle to ensure product security, usability, security, and functionality. The finished product, therefore, has far fewer problems and performs perfectly as intended, within the company's ability. ..

3. It ensures product security

Security should never be taken lightly as a lack thereof places both the company and customers at risk from nefarious entities. A QA system ensures the installation of safeguards that are built into the product or service.

4. It improves usability and accessibility

Accessibility refers to how easily different groups of people can use the same product or service while usability is about how well a software product or service can be used for specific functions with efficiency, effectiveness, and satisfaction within that context of use. QA allows the improvement of both of these components.

5. It carries out performance tests

Quality not only means failure-free software, but it also means meeting the needs of the user in a timely fashion. Performance testing helps determine the stability and responsiveness of the system under specific workloads. It also allows you to measure, verify and validate other quality features of the system.

What is software quality assurance (SQA)?

The software quality assurance (SQA) process is one that assures that all software engineering processes, activities, methods, and work items comply with defined standards, and are kept under systematic review. The standards which govern this process include one or a combination of [CMMI](#), [ISO 9000](#), [TMM](#), etc.

SOA is an umbrella activity that incorporates:

1. A quality management approach
2. Formal Technical Review
3. Multi-Testing Strategy
4. Effective software engineering techniques
5. Measurement and Reporting

There are several SQA techniques which include but are not limited to:

1. Walkthroughs: A code walkthrough or a software walkthrough is a type of peer review in which the developer shows the members of the development team how to methodically examine the product and suggest alternatives, raise queries, and make comments concerning standard violations, possible errors, or other issues.
2. Quality Audit: This is the leading technique that is widely adopted. Through inspection of the work products, this activity determines whether or not the set of standard processes was followed.

SQA Plan

The SQA Plan consists of the techniques, procedures, and tools that are used to ensure that a service or product aligns with the necessary conditions defined in the software requirement specification (SRS). The plan identifies what SQA activities a team is required to perform and makes a list of the areas requiring review and audit. SQA work products are also identified by the plan.

The following sections are included in the SQA plan:

1. Purpose
2. Reference
3. Software configuration management
4. Problem reporting and corrective action
5. Tools, technologies, and methodologies
6. Code control
7. Records: collection, maintenance, and retention
8. Testing methodology

SQA Activities

Major SQA activities include:

1. Creation of SQA Management Plan

A proper plan is developed to indicate how the SQA will be conducted on the project. It specifies what SQA approach will be followed, what engineering activities will be performed, and what mix of talent is required on your team.

2. Checkpoint Setup

Different checkpoints are set up by the SQA team where evaluations of the quality of project activities are performed at each.

3. Employment of Multi-Testing Strategy

Rather than using one testing approach, multiple types will ensure more thorough testing of the product.

4. Manage Good Relations

It is imperative that there be good relations among team members. [Effective collaboration between QA and the development team is key to product quality.](#)

5. Measure Change Impact

Defects that are reported by the QA team are fixed by the responsible team. After, the QA team should determine how the project was affected by changes made to fix this defect. They must test to ascertain that the defect was fixed and that the changes are compatible with the entire project.

It is for this reason that software quality metrics are used. They allow developers and managers to observe the activities and suggested changes from the inception of the software development lifecycle through to the end, facilitating corrective action where necessary.

When is the best time to start software QA testing?

The time and cost associated with reworking and creating error-free software can be significantly reduced by testing at the right time. The ideal time to begin testing in a software outsourcing company is at the requirements gathering stage up until deployment.

This is however contingent on the development model used. For instance, testing in an Agile model is performed at the end of each iteration and again at the end of the project, while testing in the Waterfall model is performed in the testing phase.

Different forms of testing are conducted at each stage of the software development lifecycle:

- The requirements gathering phase, verification of requirements, and analysis are all categorized as testing;
- In the design phase, issues associated with design improvement are considered part of testing;
- Testing carried out by software developers after the app's code is finished is also classified as testing.

When to stop the Software QA process

Testing may sometimes seem like a neverending process and there is no way to ascertain that software is 100% tested. Therefore, it can be difficult to decide when to stop testing.

I believe this decision depends on your budget, time, and extent of testing. However, to assure confidence in the software, you need to also take into account the number of high-priority bugs, the decrease in error rate, and the testing terms.

FAQ

What is quality assurance testing in software?

Quality assurance testing in software is the process that evaluates a software product to ensure that it meets specified requirements. It aims to improve the software development process to make it effective and efficient by identifying errors, bugs, and gaps that conflict with the requirements. Quality assurance is also generally referred to as quality assurance testing.

What is software quality assurance with example?

The software quality assurance (SQA) process is one that assures that all software engineering processes, activities, methods, and work items comply with defined standards, and are kept under systematic review. The standards which govern this process include one or a combination of [CMMI](#), [ISO 9000](#), [TMM](#), etc.

SOA is an umbrella activity that incorporates:

1. A quality management approach
2. Formal Technical Review

3. Multi-Testing Strategy
4. Effective software engineering techniques
5. Measurement and Reporting

There are several SQA techniques which include but are not limited to:

1. Walkthroughs: A walkthrough is a review technique in the static testing method. A code walkthrough or a software walkthrough is a type of peer review in which the developer shows the members of the development team how to methodically examine the product and suggest alternatives, raise queries, and make comments concerning standard violations, possible errors, or other issues.
2. Quality Audit: This is the leading technique that is widely adopted. Through inspection of the work products, this activity determines whether or not the set of standard processes was followed.

What is quality assurance in simple words?

Quality assurance is a systemic process used to determine whether a product or service meets the needs and expectations of customers.

What are the 4 steps in quality assurance?

Plan

This critical first step establishes objectives and processes that are required to ensure a high-quality end product is achieved.

Do This is the implementation phase of the plan that allows the development and testing of the quality assurance processes.

Check - This comprises the monitoring processes that allow for modification if necessary so that the predetermined objectives are met.

Act - Improvements are implemented based on the results of the monitoring process.