An Introduction To Concepts In Quality Assurance and Software Testing

Agile Assurance Group

If it’s worth building, it’s worth testing.

If it’s not worth testing, why are you wasting your time working on it?

Table of Contents:

1. Introduction

You have a product you want to launch. Is it ready? Are you confident? Are you satisfied? Is it of quality?

Software testing...is it important? Is it necessary? What does it do? What is involved?

This document addresses how Agile Assurance Group (Agile) applies Quality Assurance to all manner of software projects to greatly contribute to the quality of your product.

2. Benefits of deploying Quality Assurance (QA) Engineers

Agile QA Engineers deliver an independent and objective assessment of the software being tested. They are not only focused on ensuring the application meets the specified requirements; they also have the mindset to ensure that testing done is customer-centric.

While Software Development Engineers are driven by 'delivery', Quality Assurance Engineers are driven by 'quality'.

A product/system that undergoes testing has reduced risks compared to an untested one. It gives the stakeholders a degree of confidence.

3. Software Development Life Cycle

3.1. Involvement of Quality Assurance Engineers

The involvement of Agile's Quality Assurance (QA) Engineers in all phases of the Software Development Life Cycle (SDLC), regardless of the software development method an organization is using, ensures that continuous validation and verification are being done.

Requirements Gathering phase

In this phase, Agile's QA Engineers help in recognizing ambiguities and discrepancies in the requirements. They also verify the correctness and completeness of requirements, thus uncovering requirement-related defects, where the impact of a change required is low. They also ensure the consistency of specifications keeping the overall quality of the product in mind.

Analysis and Design phase

During this phase, Agile's QA Engineers validate that the system design addresses the features determined in the requirements document. It is also at this phase where QA Engineers start to formulate test conditions and plan how to best perform test execution.

Development and Testing Phase

Executing tests helps not only in assessing code quality but also in improving the over-all quality of the product. Through set acceptance criteria, QA Engineers validate that the system works as expected in the user or customer's point of view before being deployed. In this regard, the risk of defects being uncovered by the customer is reduced significantly.

Deployment and Maintenance phase

Any changes in a deployed system, to fix defects or implement enhancements, need to be tested. It is also necessary to conduct regression testing to ensure that unchanged parts of the system have not been adversely affected by the changes.

3.2. Agile Methodology (Scrum)

Agile Assurance Group works using the Agile methodology for most of our QA and development projects. Scrum is a framework that uses a series of sprints or iterations lasting from two to four weeks. In each sprint, the scrum team is committed to deliver a set of features or functions. At the end of each sprint, the scrum team demonstrates what has been completed to the stakeholders. Each sprint adds to the evolving product until the final product is complete.

Requirements are compiled and prioritized into product backlogs. During sprint planning, the sprint goals are determined, these goals become the deliverables. Within the sprint, the development and testing activities are performed. At the end of the sprint, a demonstration is conducted to present the completed deliverables.

QA Engineers, as part of the scrum team, participate actively in the sprint. In sprint planning, QA Engineers review the sprint scope to ensure consistency, correctness, and completeness of requirements for the sprint and at the same time ensuring that the current sprint deliverables are consistent with what has already been developed. QA Engineers not only validate the current sprint but also the entire product.

During the sprint, QA Engineers prepare and perform test activities such as creation of test cases, preparation of data and environment, perform actual test execution, verification of defect fixes, and creation of test reports. In sprint demonstration, QA Engineers often present the completed deliverables since they have a broader view of the product.

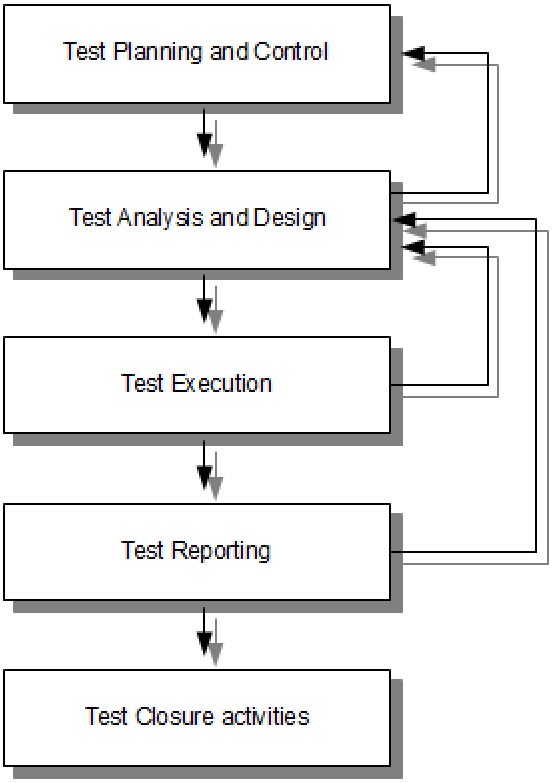
4. Software Quality Testing

4.1. Description

Mature software quality testing is not just about performing tests, more importantly it is a strict adherence to the process that goes with it. In every step of the SDLC, from business anyalysis to system maintenance, QA Engineers participate and contribute to ensure the quality of the deliverables and the product as a whole.

4.2. Quality Testing Process

This is divided into five major steps. In general, the test activities are sequential, though they interact and have an impact on each other.



Test Planning and Control

The Test Planning phase serves as a vehicle for communicating with the project team and customer the scope, approach, prioritization of areas to test, risks, resource allocation, deliverables, tasks, entry and exit criteria, and schedule. Test Control occurs throughout the duration of the project where actual progress is compared against the planned timeline and status is regularly communicated.

Test Analysis and Design

A well-designed test will uncover defects, if present, which will give more confidence in the system being built. Various test design techniques, both for functional and non-functional tests, are applied depending on the approach selected for testing.

Test Implementation and Execution

Test cases are built during this phase and any needed items, such as test data and test environment, are prepared. This is the actual test execution, where the QA Engineers test the developed system. Any defects found are then logged and fixes verified.

Test Reporting

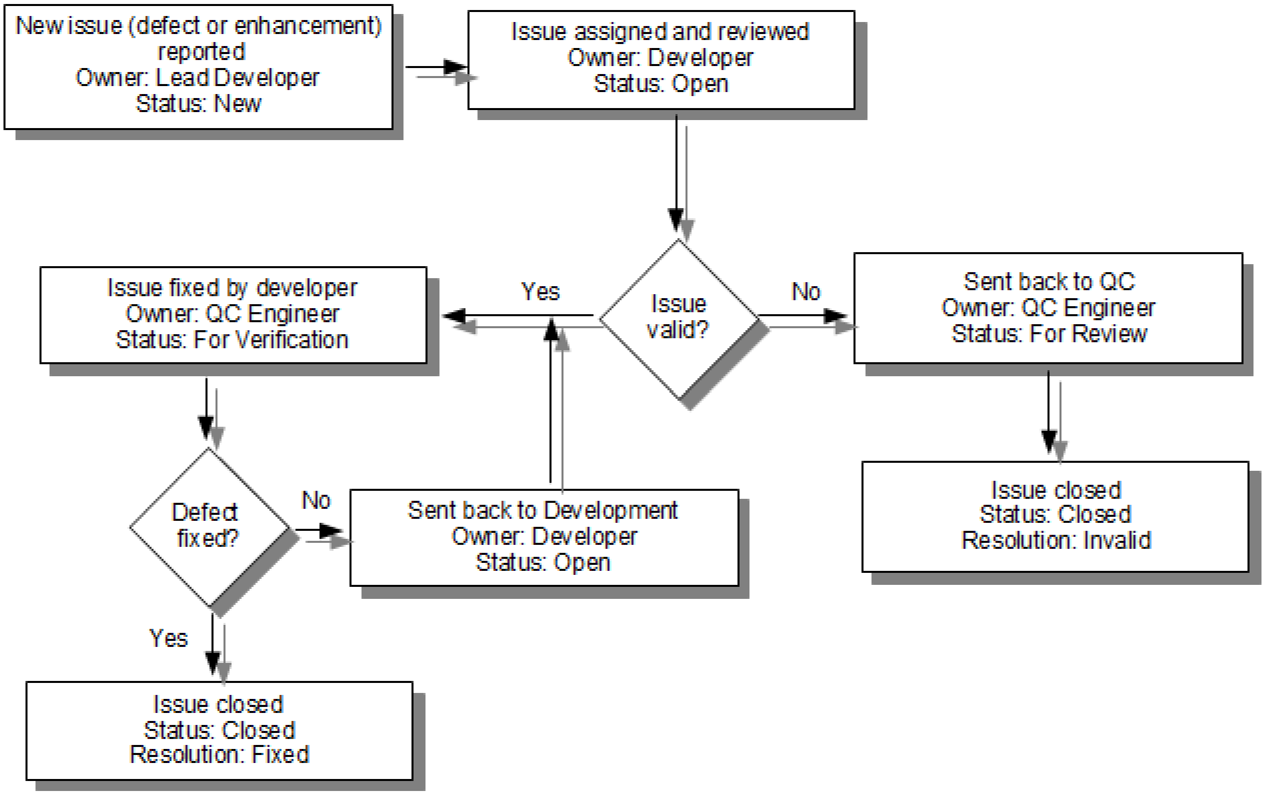
QA Engineers prepare test reports prior, during, and after the test proper. Prior to test proper, reports detail the preparations made. During test proper, the report gives the execution status and any defects logged. After the test, the report gives a summary of the testing performed.

Test Closure

After test activities are complete, the QA Engineers need to perform test closure. This involves checking what were the deliverables delivered, defects logged have been resolved or deferred, and archiving of test documents. Test closure may include regression testing on other parts of the system. Retrospective analysis is performed to evaluate how the testing went, improvements to be made, and lessons learned.

4.3. Defect Tracking Process

Discovered defects are logged and monitored in a defect management tool. The diagram displays the defect life cycle. An incident follows this cycle until it is closed.



4.4. Strategies

There are varied strategies that may be employed. Essentially, they may be grouped into three categories.

White-box testing

Testing based on an analysis of the internal structure of the component or system.

Black-box testing

Testing, either functional or non-functional, without reference to the internal structure of the component or system.

Experience-based testing

Testing based on QA Engineers' experience of a similar system and general testing experience.

4.5. Test Levels

Unit/Component Testing

Unit testing is the validation of individual software components. Tests are designed and executed by software development engineers. Proper execution of unit testing ensures that later testing phases will be more effective and discovering and fixing defects in the development stage saves project time and cost.

Integration Testing

Integration testing is performed to expose defects in the interfaces and in the interactions between integrated components or systems. Tests are designed and executed by the software development or quality control engineers.

System Testing

System testing is the process of testing an integrated system to verify that it meets specified requirements.

User Acceptance Testing

Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

4.6. Test Types

Compatibility Testing

Testing conducted on the application to evaluate the application's compatibility with the computing environment which may refer to hardware, operating systems, databases, other software, or browsers.

Confirmation Testing (Re-testing)

Testing to confirm that the fix applied for the defect has indeed fixed the problem.

Exploratory Testing

Testing that is simultaneous learning, design, and execution. Information gained as tests are performed are used to design and execute better tests.

Functional Testing

Testing based on an analysis of the specification of the functionality of a component or system. This is the most common type of testing conducted by QC Engineers. This is complemented by other types of tests.

Load Testing

The process of putting demand on a system or device and measuring its response. Load testing is performed to determine a system's behavior under both normal and anticipated peak load conditions. It helps to identify the maximum operating capacity of an application as well as any bottlenecks and determine which element is causing degradation.

Performance Testing

Testing that is conducted to determine how a system performs under any given workload. This is important in understanding system scalability, reliability, and in predicting resource usage.

Regression Testing

Testing of a previously tested program following modification to ensure that defects have not been introduced in unchanged areas of the software, as a result of the changes made.

Smoke Testing

Testing that covers the main functionality of a component or system, to ensure that the most crucial functions of a program work.

Stress Testing

Testing that is conducted to determine the upper limits of capacity within the system. This kind of test is done to determine the system's robustness in terms of extreme load and helps application administrators to determine if the system will perform sufficiently if the current load goes well above the expected maximum.

Usability Testing

Testing to determine the extent to which the software product is understood, easy to learn, easy to operate and attractive to the users under specified conditions.

User Scenario Testing

Testing that executes scenarios of use cases. Use cases are scenarios from an operational or business standpoint, how the system is to be used as defined by user.

5. Conclusion

One cannot discount the value added when testing is performed. A tested system gives a higher level of confidence, a greater degree of knowledge of what to expect, and a deep satisfaction of having a system that you desired and intended.