

## NEED OF SOFTWARE QUALITY ASSURANCE AND CONTROL IN SOFTWARE TESTING

Dr. Ravi Punia\*

Mrs. Monisha Avasthi\*

Shashiraj Teoia\*\*

### **ABSTRACT**

Software quality assurance is a planned and systematic approach to ensure that software processes and products confirms to the established standards, processes and procedures. The goals of software quality assurance are to improve software by appropriately monitoring both software and the development process to ensure full compliance with the established standards and procedures. Control of the quality of the raw material to be used in the production of pipe is the first essential and necessary condition for compliance with the specified requirements of the finished product and Software testing is oriented to "detection". It's examining a system or an application under controlled conditions. This paper describes the need of software quality assurance, quality control and testing in creation of good software.

**KEYWORDS-** Software quality assurance, Quality control, testing.

---

\* Assocaite Professor-Computer Application, ACE-Meerut

\*\* Research Scholar, IFTM University, Moradabad

## INTRODUCTION

Software is: Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system. Software quality assurance is a systematic, planned set of actions necessary to provide adequate confidence that the software development process or the maintenance process of a software system product conforms to established functional technical requirements as well as with the managerial requirements of keeping the schedule and operating within the budgetary confines and quality control activities include general methods such as accuracy checks on data acquisition and calculations and the use of approved standardized procedures for emission calculations, measurements, estimating uncertainties, archiving information and reporting. Higher tier quality control activities include technical reviews of source categories, activity and emission factor data, and methods.



### What is Quality Assurance?

Quality Assurance makes sure the project will be completed based on the previously agreed specifications, standards and functionality required without defects and possible problems. It monitors and tries to improve the development process from the beginning of the project to ensure this. It is oriented to "prevention".

### True Software QA

Software testing and quality control are essential subsets of software quality assurance. This diagram illustrates the relationships between these distinct, but interrelated levels. Transition provides comprehensive execution at every level, ensuring true Quality Assurance. Software

quality assurance is not just a service we provide; it's part of our commitment to excellence in every project we undertake. Itransition offers you the benefit of:

- ❖ Our ISO 9001:2008 certification
- ❖ The independence of our QA division from our development teams
- ❖ The maturity, transparency and continuous improvement of our QA processes,
- ❖ Our highly experienced QA staff of over 120 professionals, many of whom have applied QA knowledge management and narrow specialization
- ❖ Experience in bringing over 500 projects to successful completion
- ❖ Our depth of experience in manual and automated functional and stress testing
- ❖ Our comprehensive control over each testing stage through multiple metrics
- ❖ The integration of our processes with yours when needed

### **Why Software QA is Vital**

Integrating a solid quality assurance methodology into the larger process of software development is the best practice in every situation. Itransition never begins a project of any type without including quality assurance. Our decade-long experience proved that QA increases software quality and lowers TCO, whether for custom business solutions, software products or services such as software-as-a-service (SaaS) systems, or for an outsourced portion of a larger development effort.

**Custom Business Solutions** benefit from mitigation of risks related to misunderstanding or miscommunication of business requirements, possible system failures, and others.

**Software Products and Services** have a high degree of vulnerability to software flaws or deficiencies, because a company's future sales and ongoing revenues are directly dependent on the reliability of the product or service offered. Whether you're a startup business, an enterprise, or a product development group, quality assurance is essential to your long-term success.

**Software Development Companies and Systems Integrators** can rely on Itransition to incorporate solid QA principles into every project outsourced or assigned to us. We ensure that the modules we develop work flawlessly and satisfy every functional, technical, and architectural requirement.

### **What is Quality Control?**

*Quality Control* (QC) is a system of routine technical activities, to measure and control the quality of the inventory as it is being developed. The QC system is designed to:

- (i) Provide routine and consistent checks to ensure data integrity, correctness, and completeness;
- (ii) Identify and address errors and omissions;
- (iii) Document and archive inventory material and record all QC activities.



QC activities include general methods such as accuracy checks on data acquisition and calculations and the use of approved standardized procedures for emission calculations, measurements, estimating uncertainties, archiving information and reporting. Higher tier QC activities include technical reviews of source categories, activity and emission factor data, and methods.

### Testing Capabilities

Our software-development and QA specialists can apply a full battery of tests to software throughout the full cycle of product development as part of the Quality Control and Quality Assurance processes. These tests include:

- ❖ Functional testing
- ❖ Load and stress testing
- ❖ Regression testing
- ❖ Unit testing

- ❖ GUI testing
- ❖ Usability testing
- ❖ Security testing
- ❖ Database testing
- ❖ Cross-platform testing
- ❖ Localization testing
- ❖ Documentation testing

### **When should QA testing start in a project? Why?**

QA is involved in the project from the beginning. This helps the teams communicate and understand the problems and concerns, also gives time to set up the testing environment and configuration. On the other hand, actual testing starts after the test plans are written, reviewed and approved based on the design documentation.

### **What is Software Testing?**

Software testing is oriented to "detection". It's examining a system or an application under controlled conditions. It's intentionally making things go wrong when they should not and things happen when they should not.

### **What is Software Quality?**

Quality software is reasonably bug free, delivered on time and within budget, meets requirements and/or expectations, and is maintainable.

### **What is Software Verification and Validation?**

Verification is preventing mechanism to detect possible failures before the testing begin. It involves reviews, meetings, evaluating documents, plans, code, inspections, specifications etc. Validation occurs after verification and it's the actual testing to find defects against the functionality or the specifications.

### **What is Test Plan?**

Test Plan is a document that describes the objectives, scope, approach, and focus of a software testing effort.

### **What is Test Case?**

A test case is a document that describes an input, action, or event and an expected response, to determine if a feature of an application is working correctly. A test case should contain

particulars such as test case identifier, test case name, objective, test conditions/setup, input data requirements, steps, and expected results.

### **What is Good Software Coding?**

Good code is code that works according to the requirements, bug free, readable, and expandable in the future and easily maintainable.

### **What is a Good Design?**

In good design, the overall structure is clear, understandable, easily modifiable, and maintainable. Works correctly when implemented and functionality can be traced back to customer and end user requirements.

### **Who is a Good Test Engineer?**

Good test engineer has the ability to think the unthinkable, has the test to break attitude, strong desire to quality and attention to detail.

### **What is Walkthrough?**

Walkthrough is quick and informal meeting for evaluation purposes.

### **What is Software Life Cycle?**

The Software Life Cycle begins when an application is first conceived and ends when it is no longer in use. It includes aspects such as initial concept, requirements analysis, functional design, internal design, documentation planning, test planning, coding, document preparation, integration, testing, maintenance, updates, retesting, phase-out, and other aspects.

### **What is Software Inspection?**

The purpose of inspection is trying to find defects and problems mostly in documents such as test plans, specifications, test cases, coding etc. It helps to find the problems and report it but not to fix it. It is one of the most cost effective methods of software quality. Many people can join the inspections but normally one moderator, one reader and one note taker are mandatory.

### **What are the benefits of Automated Testing?**

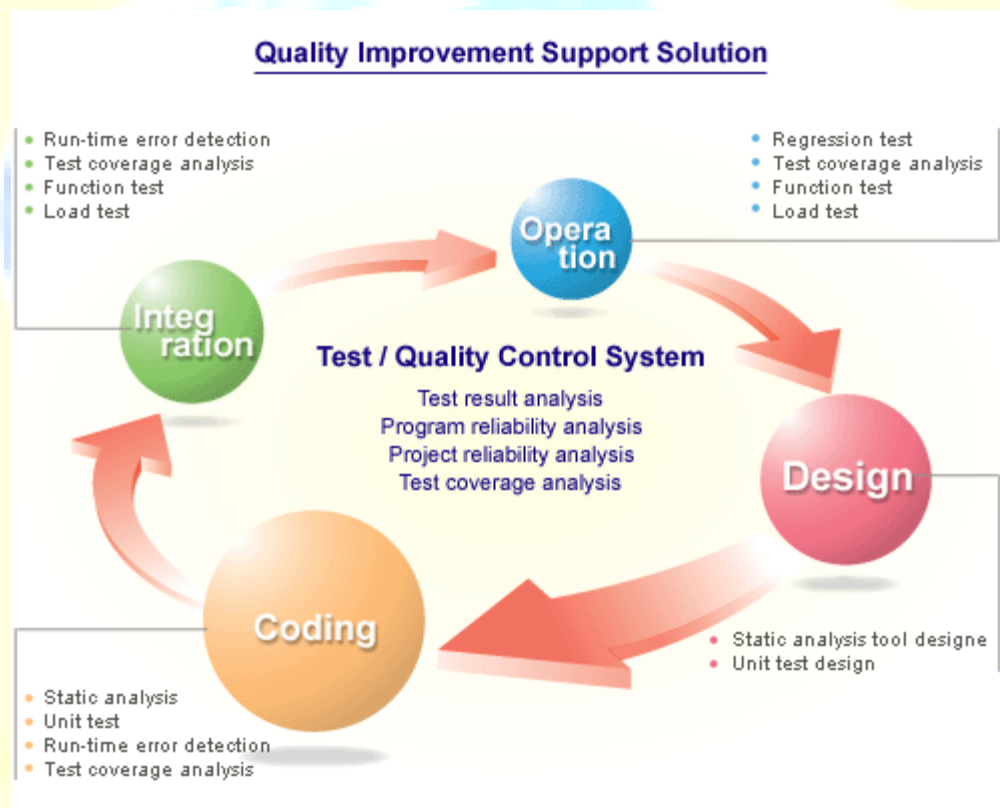
It's very valuable for long term and on going projects. You can atomize some or all of the tests which needs to be run from time to time repeatedly or difficult to test manually. It saves time and effort, also makes testing possible out of working hours and nights. They can be used by different people and many times in the future. By this way, you also standardize the testing process and you can depend on the results.

**What do you imagine are the main problems of working in a geographically distributed team?**

The main problem is the communication. To know the team members, sharing as much information as possible whenever you need is very valuable to solve the problems and concerns. On the other hand, increasing the wired communication as much as possible, setting up meetings help to reduce the miscommunication problems.

**What are the common problems in Software Development Process?**

Poor requirements, unrealistic schedule, inadequate testing, miscommunication and additional requirement changes after development begin.



**What are Software Testing Types?**

- ❖ **Black box testing:** You don't need to know the internal design in detail or have a good knowledge about the code for this test. It's mainly based on functionality and specifications, requirements.
- ❖ **White box testing:** This test is based on detailed knowledge of the internal design and code. Tests are performed for specific code statements and coding styles.

- ❖ **Unit testing:** The most micro scale of testing to test specific functions or code modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. Not always easily done unless the application has a well-designed architecture with tight code, may require developing test driver modules or test harnesses.
- ❖ **Incremental integration testing:** Continuous testing of an application as new functionality is added. Requires that various aspects of an application's functionality be independent enough to work separately before all parts of the program are completed, or that test drivers be developed as needed. Done by programmers or by testers.
- ❖ **Integration testing:** Testing of combined parts of an application to determine if they function together correctly. It can be any type of application which has several independent sub applications, modules.
- ❖ **Functional testing:** Black box type testing to test the functional requirements of an application. Typically done by software testers but software programmers should also check if their code works before releasing it.
- ❖ **System testing:** Black box type testing that is based on overall requirements specifications. Covers all combined parts of a system.
- ❖ **End to End testing:** It's similar to system testing. Involves testing of a complete application environment similar to real world use. May require interacting with a database, using network communications, or interacting with other hardware, applications, or systems.
- ❖ **Sanity testing or smoke testing:** An initial testing effort to determine if a new s/w version is performing well enough to start for a major software testing. For example, if the new software is crashing frequently or corrupting databases then it is not a good idea to start testing before all these problems are solved first.
- ❖ **Regression testing:** Re-testing after software is updated to fix some problems. The challenge might be to determine what needs to be tested, and all the interactions of the functions, especially near the end of the software cycle. Automated testing can be useful for this type of testing.
- ❖ **Acceptance testing:** This is the final testing done based on the agreements with the customer.



- ❖ **Load / stress / performance testing:** Testing an application under heavy loads. Such as simulating a very heavy traffic condition in a voice or data network, or a web site to determine at what point the system start causing problems or fails.
- ❖ **Usability testing:** Testing to determine how user friendly the application is. It depends on the end user or customer. User interviews, surveys, video recording of user sessions, and other techniques can be used. Programmers and testers are usually not appropriate as usability testers.
- ❖ **Install / Uninstall testing:** Testing of full, partial, or upgrade install / uninstall processes.
- ❖ **Recovery / failover testing:** Testing to determine how well a system recovers from crashes, failures, or other major problems.
- ❖ **Security testing:** Testing to determine how well the system protects itself against unauthorized internal or external access and intentional damage. May require sophisticated testing techniques.
- ❖ **Compatibility testing:** Testing how well software performs in different environments. Particular hardware, software, operating system, network environment etc. Like testing a web site in different browsers and browser versions.
- ❖ **Exploratory testing:** Often taken to mean a creative, informal software test that is not based on formal test plans or test cases; testers may be learning the software as they test it.
- ❖ **Ad-hoc testing:** Similar to exploratory testing, but often taken to mean that the testers have significant understanding of the software before testing it.
- ❖ **Context driven testing:** Testing driven by an understanding of the environment, culture, and intended use of software. For example, the testing approach for life critical medical equipment software would be completely different than that for a low cost computer game.
- ❖ **Comparison testing:** Comparing software weaknesses and strengths to competing products.
- ❖ **Alpha testing:** Testing of an application when development is nearing completion. Minor design changes may still be made as a result of such testing. Typically done by end users or others, not by programmers or testers.

- ❖ **Beta testing:** Testing when development and testing are essentially completed and final bugs and problems need to be found before final release. Typically done by end users or others, not by programmers or testers.
- ❖ **Mutation testing:** A method for determining if a set of test data or test cases is useful, by deliberately introducing various code changes (defects) and retesting with the original test data/cases to determine if the defects are detected. Proper implementation requires large computational resources.

### Software Testing and QA

Quality Assurance department is ISO 9001:2008 certified and comprises 100+ experienced QA engineers dedicated solely to web and applications testing as well as testing processes automation. Testing processes are based on Rational Unified Process (RUP) best practices and Agile development methodologies – a mix that ensures highest quality work is performed every time, thus shortening the time-to-market for software products and ensuring product are deployed in the most efficient way on the customer's side.

**Independent testing service offering** by Itransition covers 3 major areas:

- Software testing services (web and systems applications)
- Embedded systems testing
- Automated tests development

Independent Quality Assurance allows for a better control of the quality of the application, makes the evaluation of the product's compliance with the original requirements fast and easy and provides an insight to the product's convenience to the end-user.

We offer the following testing and quality assurance services:

- ❖ Software testing
- ❖ Installation testing
- ❖ Functional testing
- ❖ Automated testing
- ❖ Technical testing
- ❖ Web applications testing
- ❖ Security testing
- ❖ Testing of graphical user interface

- ❖ Compliance Testing (508, UK and European accessibility standard, etc.)
- ❖ Quality control (Quality assessment/ Quality monitoring)
- ❖ Documentation Services (Technical writing/ Proofreading and editing / Technical documentation and software localization).

You don't have to wait until the development process is finished to start testing your solution. The quality assurance process should occur simultaneously with the development process, and the system can and should be tested before its final implementation. This will save considerable amounts (up to 60%) of money and time needed for debugging and what is even more important it will enhance your reputation and improve the overall user satisfaction with your software product.

## CONCLUSION

Quality is a complex concept. Because it means different things to different people, it is highly context-dependent. Just as there is no one automobile to satisfy everyone's needs, so too there is no universal definition of quality. To assess or improve software quality in your organization, you must define the aspects of quality in which you are interested, and then decide how you are going to measure them. By defining quality in a measurable way, you make it easier for other people to understand your viewpoint and relate your notions to their own. Ultimately, your notion of quality must be related to your business goals. Only you can determine if good software is good business.

## REFERENCES

1. R. Howe, D. Gaeddert, and M. Howe, *Quality on Trial*, McGraw-Hill Europe, Maidenhead, England, 1992.
2. D. Garvin, "What Does "Product Quality" Really Mean?" *Sloan Management Review*, fall 1984, pp. 25-45.
3. *ISO 9001 Quality Systems - Model for Quality Assurance in Design/Development, Production, Installation, and Servicing*, International Organization for Standardization, Geneva, 1994.
4. Special issue on reliability measurement, *IEEE Software*, July 1992.
5. *ISO 9001 Quality Systems Model for Quality Assurance in Design/Development, Production, Installation, and Servicing*, International Organization for Standardization, Geneva, 1994.
6. Prof. Je\_ Tian, SMU, S/w Quality Assurance, Nov. 30, 2002
7. IEEE Guide for software quality assurance planning, ANSI/IEEE Std 983-1986