

---

**THE ROLE OF SOFTWARE TESTING IN SQA, MANUAL VS  
AUTOMATION: A CASE BASED ANALYSIS**

Taruna\*

Dr. Chanderkant\*\*

---

**ABSTRACT**

*The importance of Software Quality Assurance (SQA) has been known from the advent of software engineering; however the practical implementation of it has been observed growing rapidly due to experienced software crises. Software Quality Assurance (SQA) is generally oriented to 'prevention'. It involves the entire software development process. Software Testing is also oriented to 'detection'. Testing involves the operation of a system or application under controlled conditions and evaluating the results. This paper analysis the role of software testing in SQA process. Also provide a comparison between manual and automated testing approaches. This case study based on experiences shared by some software testing people (QA) who are working for different software organizations.*

**Key Words:** *Software Quality Assurance (SQA), Testing Tools*

---

\*Research Scholar, Department of Computer Science, Monad University, Hapur (U.P.)

\*\*Assistant Professor, Deptt. of Comp. Sc. & App., Kurukshetra University, Kurukshetra (HR)

## 1. INTRODUCTION

Software quality initiative is twofold, involving software quality assurance (SQA) and software quality control. However, the difference between the two is more than semantics. Quality control is an activity that ensures that the product or service meets the company's defined quality standards. It is commonly known as "testing."

Software QA involves the entire software development PROCESS - monitoring and improving the process, making 'sure that any agreed-upon standards and procedures are followed, and ensuring that problems are found and dealt with. It is oriented to 'prevention' [6].

Testing involves operation of a system or application under controlled conditions and evaluating the results (e.g., 'if the user is in interface A of the application while using hardware B, and does C, then D should happen'). The controlled conditions should include both normal and abnormal conditions. Testing should intentionally attempt to make things go wrong to determine if things happen when they shouldn't or things don't happen when they should. It is oriented to 'detection' [6].

## 2. NEED FOR SQA

There are so many reasons why a company should consider SQA. It is all about business survival. The ultimate goal of SQA is to determine quality product. Quality of software is defined as its ability to satisfy all implied needs. SQA is a set of assurance processes that are implemented across all phases of the project and software development lifecycles [1].

The activities in the SQA process may include formal reviews, testing, control of documentation, measurement, and reporting (Pressman, 2005). At a high level, the function of SQA is to perform the following [4]:

- **Assure software project planning has taken place:** Quality practices should be planned well in advance so that there is time for them to be implemented.
- **Assure user requirements:** Requirements should be reviewed from beginning to end to assure conformance to established standards and conformance to user needs.
- **Assure the design process:** Provide guarantee that methodologies are followed and requirements are met by the design.

- 
- **Assure coding practices:** Coding standards, practices, and guidelines must first be established, and then adhered to.
  - **Assure software integration and testing has taken place:** Software integration and testing should be planned, implemented, and executed accordingly.
  - **Conduct random and scheduled audits:** Perform SQA audits to assure the necessary controls are in place [4].

The overall goal of .QA is to deliver software that minimizes defects and meets specified levels of function, reliability, and performance. 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 [10].

### 3. TESTING GENERAL ISSUES

Testing is a collection of techniques used to measure, and thereby improves, software quality. Testing fits in the broader category of software management practices known as quality assurance (QA), along with other practices, such as defect tracking and design and code inspections [5].

A common perception of testing is that it only consists of running tests, i.e. executing the software. This is part of testing, but not all of the testing activities.

Test activities exist before and after test execution, activities such as planning and control, choosing test conditions, designing test cases and checking results, evaluating completion criteria, reporting on the testing process and system under test, and finalizing or closure (e.g. after a test phase has been completed). Testing also includes reviewing of documents (including source code) and static analysis [5].

Both dynamic testing and static testing can be used as a means for achieving similar objectives, and will provide information in order to improve both the system to be tested, and the development and testing processes.

There can be different test objectives:

- Finding defects;
- Gaining confidence about the level of quality and providing information;
- Preventing defects.

Software testing in many software industries carried out in two major ways, manual and

automation [6]. In manual process testers (QA) keep writing test cases based on software requirement specification and execute those test cases against actual software application being developed. In automated process different automation tools are used for testing an application. In automated process testers need to write scripts which are then executed with help of testing tools.

The survey conducted shows that most of the testing people are not satisfied with the manual as well as automated process they are using to test an application.

So based on the experiences shared by some software industry people (QA), we have summarized the following observations:

#### **About Manual Process**

- Working in agile fashion and use to develop and test the product in sprints/iterations. At the end this leads to pile up the QA work and when project completion date comes closer QA get overloaded, QA get tired and level of quality could not be achieved.
- Due to limited rights testers can do the things that as what they wanted.
- Sometimes it's irritating to do the things manually as it could be added to testing tools they are using.
- Most tedious job in manual process is to write test cases and update them according to actual results.
- It is time consuming activity. It is so hard to file a bug with exact step that hit the issue.
- The bug that tester file sometimes are not reproducible on the developers machine, due to differences in configuration / environment.

#### **About Automated Process**

- Testing tools are really an excellent alternative for manual testing, but there are certain which are out of scope for Automation. For things example, integrating certain modules and evaluating the results of Integration may be done with Automation, but may not result in completely reliable integrated module.
- You may have missed certain loop from the original code while you are writing automation code. These things need to be checked manually only to avoid any loop-

---

holes created during Development. Having said that, it's worth to notice, Automation has caused a big impact on overall testing process, so it still have an upper hand over Manual Testing.

#### **4. DEFINING A SCOPE FOR TEST PROCESS IMPROVEMENT**

Testing to be broken in to projects and test team members to be assigned projects on an experience basis. Test automation to be adopted and the training of the best practices and use of the tool to be implemented.

Bug tracking, Bugzilla database to be backed up on a daily basis. Procedure for the tracking of bugs using Bugzilla to be implemented.

The testing of a software application to be divided in to three sections/iterations:

- The first iteration is to automate the tests and to identify as many tickets as possible.
- The second is to regression test the bug fixes and improve the scripts where necessary. Any tickets found will be documented and metrics calculated.
- The third regression test will be to verify that the application meets release criteria.

After each test section the application will be returned to the developer for the resolution of any tickets.

System testing to be implemented for the release of a CD or a new application which is to be added to the suite. An improved test area to be set up and documented for the system test. Should test each application for functionality and performance. Firmware testing to be addressed, the feasibility of automating the firmware testing using a developed COM interface or other testing tools .A tester should be product expert. Certain heuristics can be used which will give you an idea where more errors will be.

#### **5. CONCLUSION**

From above discussion we can say that software testing plays a vital role in software quality assurance (SQA) process. Again we come to know that both manual and automated approaches of software testing in SQA process are equally important. Both testing are like two horses in a chariot we can even say manual testing and automation testing are two phases of testing once manual is over then they will go for automation and it depends upon the project too.

In summary, we would say automation is for repeated and generic functionality that does

---

---

not change on regular basis. Manual test is for more complex scenarios / cases using risk based approach.

## REFERENCE

1. Feldman, S. (2006), Quality Assurance: Much More than Testing. Retrieved February 10, 2008, from Association from Association for Computing Machinery.
2. Software Quality Assurance, Dawn M. Owens University of Nebraska at Omaha, USA. Deepak Khajanchi University of Nebraska at Omaha, USA. .
3. Software Testing and Quality Assurance White Papers, Pointe Technology Group, Inc.
4. Software quality assurance (2009), Daniel Galin, Thomas Publication Inc., USA.
5. Testing: First step towards software quality (2007), Mihnea Iacob, Radu Publications, Korea.
6. [www.Software QATest.com](http://www.SoftwareQATest.com)
7. Guide to software quality assurance (2010), ESA Board for Software Standardization and Control, ANC, London.
8. Incorporating Software Quality Assurance in Computer Science Education: An Experiment, Massood Towhidnejadi, November 6-9, 2002, Boston, MA 32nd ASEE, IEEE Frontiers in Education Conference.
9. Quality Indicators for Software Analysis, Fadi T.R., Alheresh and Kashif Riaz, 5th Saudi Technical Conference and Exhibition, Riyadh, 11-14 Jan. 2009
10. Adopting SQA Practices for Defect Prevention, [www.keane.com](http://www.keane.com)