

A Comparative Analysis Study Of Software Testing Tools With Quality Factors Affects In Software Development Cycle

Dr. Balasaheb N. Bhamangol

D. Y. Patil Institute of MCA and Management

Abstract

Software testing is very important phase in software development life cycle (SDLC), to test software we need to use software testing life cycle (STLC). In early days IT industries do apply manual testing strategy for testing software but now days they do it by automation process also. To automate testing process various tools are required. Test automation tools enable developers and testers easily automate the entire process of testing in software development. The researcher has focused the comparative study of automated testing tools such as the Quick Test Professional (QTP), Load Runner, Win Runner and JUnit based on quality parameters. The fundamental goal is to analyze the effectiveness and efficiency of tools for software testing by these tools with the quantitative quality parameters.

Keywords: - Software Development Life Cycle, Software Testing Life Cycle, LoadRunner, Quick Test Pro, WinRunner, Junit

1. Introduction

1.1 System Development Life Cycle: The Systems development life cycle (SDLC), or Software development process in software engineering, is a process of creating or altering information systems, and the models and methodologies that people use to develop these systems. Software development process uses different phases such as requirement gathering, Analysis, design, coding, testing and implementation.

- **Requirement:** It is used to plan the basic project approach and to conduct product feasibility study in the economical, operational, and technical areas.
- **Design:** System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
- **Development:** On receiving system design documents, the work is divided in modules/units and actual coding is started.
- **Testing:** After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase.
- **Implementation:** Modular and subsystem programming code will be accomplished during this stage.

1.2. V & V Model (Verification & Validation):

V and V model stands for verification and validation. In diagram software development stages and software testing phases have been shown. Software development starts with requirement gathering, analysis, design, coding build software, build system and implementation. Software testing starts with review of requirement gathering and simultaneously preparation of test plan, analysis review, in this phase revised test plan takes place where changes in requirement, suggestions from customers are consider and then according to the changes test plan get revised. Design review describes verification of architecture, diagrams, screen shots/forms, reports. Code review generally performs by developer which is also called unit testing. Integration testing performs against group of components, modules. System testing performs against complete development of software; here we can apply different types of testing like load, stress, performance, usability, compatibility, security etc. At last acceptance testing involves user.

Fig 2. V & V Model

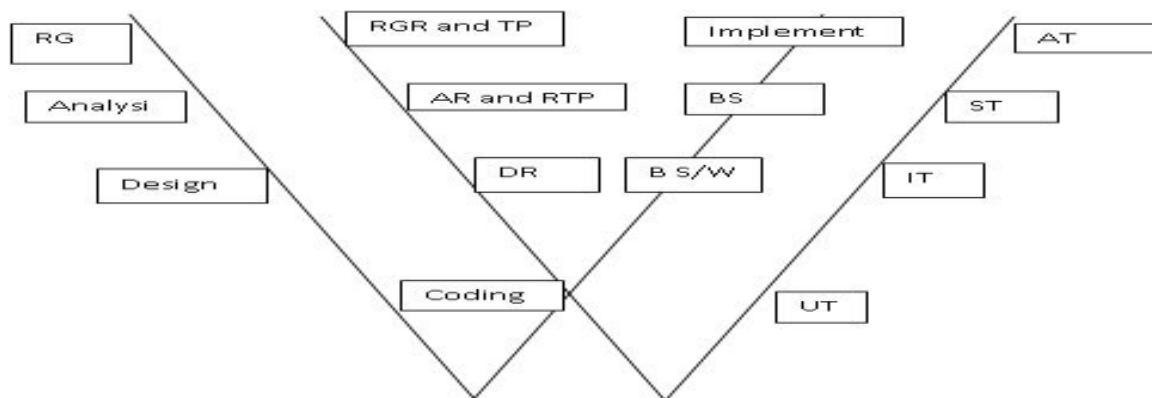


Fig. V & V Model

RG – Requirement Gathering, DR – design Review

AR and RTP – Analysis Review and Revised Test Plan

RGR and TP - Requirement Gathering and Test Plan Preparation

B S/W – Build Software, BS – Build system, UT – Unit testing

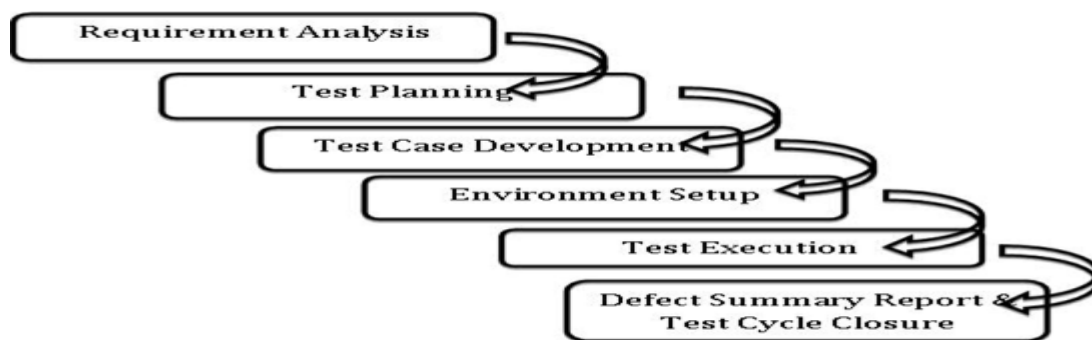
IT – Integration Testing, ST – System Testing, AT – Acceptance Testing

1.3 Software Testing Life Cycle

- **Requirement Analysis:** In this step Quality Assurance (QA) team understands the requirement in terms of what we will testing & figure out the testable requirements.
- **Test Planning:** In this phase typically Test Manager involved to determine the effort and cost estimates for entire project.
- **Test case Development:** Test case development is phase of STLC where testing team write down the detailed test cases. Along with test cases testing team also prepare the test data if any required for testing.
- **Environment setup:** Test environment decides on which conditions software is tested.

- **Test Execution:** In this phase testing team start executing test cases based on prepared test planning & prepared test cases in the prior step.
- **Defect Summary Report & Test Cycle Closure:** Final defect report is prepared based on no. of defects found in the software by the tester with using defect report summary fields like defect id, test case id, status.

Fig 3. Software Testing Life Cycle



1.4 Quality Parameters used for Testing

- **Software Quality:** Software quality is the characteristic of the software that defines how well the software meets the customer requirements, business requirements, coding standards etc. It can be divided into two categories:
- **Software Functional Quality:** characteristics that define how well the software meets functional requirements, and how well it satisfies the end-users.
- **Software Non-Functional Quality:** characteristics that define how well structural requirements are met that support the delivery of functional requirements. It is usually related to software code and internal structure.

The different software qualities can be measured through various software testing techniques and tools. Following are the different attributes (parameters) that are used to measure the software quality:

- 1) **Testability** – How easy it is to test the software and to what extent it can be tested.
- 2) **Usability** – It defines how user friendly the software is.
- 3) **Understandability** – How easily the software can be made understood to a layman about its functions/purpose
- 4) **Consistency** – How far the software is consistent / uniform in terms of GUI, terminologies, conventions.
- 5) **Efficiency** – It defines the amount of work that the software can do in defined timeframe with minimum resources used.
- 6) **Effectiveness** – The extent to which the software satisfies the user and meets user requirements
- 7) **Accuracy** – How accurately the software works with gives the correct results.
- 8) **Maintainability** – How easily the software can be maintained in terms of enhancements/new features/bug fixes.
- 9) **Reliability** – How reliable the software is in performing required functions under

different scenarios/conditions.

- 10) **Portability** – How easily the software can be transported and run in different environments e.g. platforms like operating systems (Linux, Mac, Windows), machines it can run on.
- 11) **Security** – How secured the software is in terms of access authorization and personal data like passwords.
- 12) **Robustness** – How robust the software is under unexpected events like software crash, power-off etc and saves its data.

2. Review of Literature:

Review 1: A STUDY ON QUALITY PARAMETERS OF SOFTWARE AND THE MATRICE FOR EVALUATION, J.Emi Retna, Greeshma Varghese, Marlin Soosaiya, Sumy Joseph, Karunya University, Coimbatore.

Software quality is one of the illusive targets to achieve in the software development. The successful software projects achieve excellence in software quality. Today it is viewed as an essential parameter with any software delivered. Software quality has several definitions and is viewed in several perspectives. Software built with all the requirements specified by the client or software that has 0 % defect cannot be deemed as of high quality. Quality however is not a single parameter but it is a collection of parameter and it has a multidimensional concept.[1]

Review 2: Best Software Test & Quality Assurance Practices in the project Life cycle, Mark Kevitt, BSc, Dublin City University April 2008

A software company with software quality problems This thesis is focused on the creation and provision of a testing & quality assurance process for software quality improvement in as Irish company and also for the creation of a framework for similar quality improvement in the process for other companies.

This is a product of the research into test & QA(Quality Assurance) practices and for the provision of an improved test process in the company. This process will combine element of testing and QA into one process, this one process in turn will be inserted into the company's Life Cycle.[2]

Review 3: Identification & Analysis of Parameters for Program Quality Improvement: A reengineering Perspective, HarmeetKaur, Punjab Technical University Jalandhar, Punjab, ShahanawajAhamad, Dept. of C.S. & Soft. Engg., College of comp. Sc. & Engg., University of Ha'il, K. S. A., Gurvinder N. Verma, ShriSukhmani Institute of Engg. & tech., Derabasi, Punjab.

The rise in demand for efficient systems has increased greatly in last few years, but there has been also criticism about the quality of the software system being currently used. As the expectations for quality have raised, it is insufficient to deliver the software to product which is technically sound but does not meet the end users expectations i.e. easy to use and it should fit best in the work practices and activities of the end user. The mindset is to remember that a software product is developed to provide a range of services for a user group, in order to help them achieve certain needs or goals. Thus, it should be clear that for any software project one

should precisely know what those needs or goals are. These are the key drivers behind the identification of not just the functional requirements, but also the quality requirements.[3]

Review 4: Importance of Testing in Software Development Life Cycle, T. Rajani Devi.

Software development life cycle is working like or equal to the layer number two i.e. Data Link Layer. It decides the proper flow of the data from one level to another level. In every mainframe network, host mainframe is considered as primary workstation and another device are known as secondary. As compare with SDLC, SDLC uses primary workstation and secondary workstation for its communication mode or medium. Also it is true that every secondary devices or workstation has its own address and they are linked with common ports which are called as multi-point arrangement. From this point of view, SDLC used for point-to-point communication. Research said that the test process frequently accounts for 40 % of the cost of software development. According to author, there are many important things associated with SDLC. It is compared as the basis of Standard Data Link Protocol in ISO, High Level Data Link Control. SDLC are very effective & efficient protocol and it is used in close network with its own private lines. At last the very important thing is the SDLC phases defines the key schedule and correct delivery to the client within the budget and other project requirement.[4]

Review 5: Comparative Study of Automated Testing Tools:Quick Test Pro and Load Runner, Shaveta, Sachinkumar, Nitika, SnehlataC.S.E student, Department Of Computer Science Engineering, Punjab.

The software testing process is held to identify defects in existing system or software product. It is process to check the fulfillment of user's requirements by manual testing as well as automotive testing and evaluating the system or system components. And get the differences between expected and actual result.

There are mainly two types of testing: one is Manual and second is Automation. Where as in automation testing, software testing is done by using some testing tools. And manual testing carried out by the testers. Tester is the person who is test the software to get the error or bugs of the system. It requires a tester to play the role of an end user. They follow a written test plan that leads them through a set of important test cases.[5]

Review 6: A SYSTEMATIC REVIEW OF AUTOMATED SOFTWARE TESTING TOOLS, A. NIRMAL KUMAR, DR. B. G. GEETHA

Manual testing is time consuming process. The automated testing speed up the testing process. Manual testing requires a heavy investment in human resources. In automated testing, we can create test scripts that check all aspects of application. The benefits of automated testing are fast, reliable, comprehensive and reusable. The tools which are used only for testing is call as an automated test tools. The tools are classified into functional test tools, performance test tools, test management tools. The functional test tools are used for checking only the functionality of the software. The performance test tools are used for checking the performance of the software and management tools are used to manage the entire testing activity like creating test plan, test case design, defect tracking.[6]

Review 7: Importance of Software Testing in Software Development Life Cycle :Mohd. Ehmer Khan, Dept. of Information Technology, Al Musanna College of Technology,

Sultanate of Oman. Fermeena Khan, Dept. of Computer Science, EIILM University, Jorethang, Sikkim, India.

Software development life cycle is a structure imposed on the development of a software product. There are different activities involved in SDLC such as requirement, specification, architecture, software construction, design, software testing, debugging, deployment and maintenance. Software testing plays a vital role in each and every phase of SDLC. The goal of application performance testing is to appraise any user experience in realistic scenarios on our target application. In software engineering, performance testing is in general testing performed to determine how a system performs in terms of responsiveness and stability under every condition.[7]

3. Research Methodology

Present study is comparison of various testing tools like Quick Test Pro (QTP), LoadRunner, Winrunner & Junit with quality parameters, so researcher has referred existing Literatures from various journals as a secondary data.

4. Data Representation, Analysis and Interpretation

For this research we use the Quick Test Pro (QTP), LoadRunner, Winrunner & Junit testing tools for comparison and identify the results of record efficiency, Capability of generation of script, Data Driven Testing, Test result report. Also test is applied on various general parameters that are mainly focuses on usability, security, efficiency, accuracy, reliability and robustness. For analysis purpose of we used scaling technique like 1 indicates Bad followed by 2 indicates Average whereas 3 indicates Good further 4 indicates Very Good, and 5 indicates Excellent.

4.1 Recording Efficiency: For comparison of recording efficiency of various parameters recording type, insert command, Access to record control and Auto documentation are considered with respect to various testing tools like QTP, Load Runner, Win Runner & Junit. For measuring recording efficiency various parameters are considered like Recording type, Access to record control and Auto documentation. As per the following Table No.4, it clears that QTP & Load Runner has excellent recording efficiency quality that is 5 followed by WinRunner has Good quality that is 3. All tools have very bad quality of record efficiency with respect to Insert command parameter. QTP, LoadRunner and WinRunner gives excellent auto Documentation quality that is 5 followed by Junit has Very Good that is 4.

Table No. 1: Record Efficiency

Recording Efficiency				
Tools → Parameters ↓	QTP	Load Runner	WinRunner	Junit
Recording type	5	5	3	-
Insert command	1	1	1	1
Access to record control	1	1	2	4
Auto documentation	5	5	5	4
Calculation	For QTP The value of parameter is $=1+5+1+5/4=12/4=3$ For Load Runner The value of parameter is $=1+5+1+5/5=12/5=3$ For WinRunner The value of parameter is $=3+1+2+5/4=11/4=2.75$ For Junit The value of parameter is $=1+4+4/3=9/3=3$			

(Note: 1- Bad , 2 - Average, 3-Good, 4-Very Good, 5-Excellent)

As per the table QTP, Load Runner, Junit have a Good record efficiency that is 3 and WinRunner has also good record efficiency with value 2.75. It clears that recording efficiency of QTP, Load Runner, Junit is Good as compared to WinRunner.

4.2 Capability of Generation of Script: For measuring capability of testing tools language as parameter considered. Following Table No. 2 shows the Capability of generation of script. As per the table capability of Junit is 5 as compared to QTP is 4 followed by Load Runner and Win Runner has 1.

Table No. 2: Capability of generation of script

Capability of Generation of Script				
Tools → Parameters ↓	QTP	Load Runner	WinRunner	Junit
Language	4	1	1	5

(Note: 1- Bad, 2 - Average, 3-Good, 4-Very Good, 5-Excellent)

Junit has Excellent Capability of generation of script followed by QTP has Very good capability whereas Load Runner and Win Runner have very bad capability.

4.3 Data Driven Testing: For data driven testing, Access data from external source, Change the data without effecting script, Way of testing parameters are considered. Following table No. 3 shows the result of Data Driven Testing.

For measuring quality of data driven testing various parameters are considered like Access data from external source, Change the data without effecting script and Way of testing. As per the following Table No.3, it clears that QTP, Load Runner & Win Runner has excellent (5) quality of Access data from external source as compared to Junit. QTP and Load Runner has excellent (5) quality to change the data during testing without effecting script followed by WinRunner has a very Good (4) quality to change the data during testing without effecting script.

Table No. 3: Data Driven Testing

Data Driven Testing				
Tools -> Parameters ↓	QTP	Load Runner	WinRunner	Junit
Access data from external source	5	5	5	-
Change the data without effecting script	5	5	4	-
Way of testing	5	4	4	3
Calculation	For QTP The value of parameter= $5+5+5=15/3=5$ For Load Runner The value of parameter= $5+5+4/3=14/3=4.6$ For WinRunner The value of DDT is= $4+4+5/3=13/3=4.3$ For Junit The value of DDT is= $3/3=1/3= 0.33$			

(Note: 1- Bad, 2 - Average, 3-Good, 4-Very Good, 5-Excellent)

It clears that QTP is excellent data driven testing quality followed by Load Runner & WinRunner have Very Good and Junit is Very Bad.

4.4 Test Report: As a result of testing and for a Test Report various parameters are considered like Report Presentation, Information about applied check point, Graphical information. As per the following Table No.4, it clears that QTP has excellent presentation quality that is 5 followed by Load Runner & WinRunner have Very Good quality that is 4 and Junit have a Good quality that is 3. QTP and LoadRunner gives excellent information about applied check point that is 5 as compared to WinRunner has Very Good that is 4. Also all the tools are excellent in generation of graphical information that is 5.

Table No.4: Test Report

Test Result Report				
Tools -> Parameters ↓	Q TP	Load Runner	WinRunner	Junit
Report Presentation	5	4	4	3
Information about applied check point	5	5	4	-
Graphical information	5	5	5	5
Calculation	For Q TP The value of parameter= $5+5+5=15/3=5$ For Load Runner The value of parameter= $5+5+4/3=14/3=4.6$ For WinRunner The value of DDT is= $4+4+5/3=13/3=4.3$ For Junit The value of DDT is= $5+3/3=8/3=2.6$			

(Note: 1- Bad , 2 - Average, 3-Good, 4-Very Good, 5-Excellent)

As per above table and calculations, it clears that QTP is Excellent tool (5) regarding testing result report as compared to the other tools like Load Runner (4.6) followed by WinRunner (4.3) and further Junit (2.6).

Hence it clears that QTP is a excellent tool for generating test result as compared the other tools like Load Runner, WinRunner & Junit

4.5 Quality Parameters: To do the comparative study of various testing tools, various quality parameters are considered like Usability, Security, Efficiency, Accuracy, Reliability & Robustness. Following table No. 5 shows the different parameters that are used to measure the software quality by using various tools

Table No. 5: Quality Parameters

Quality parameters				
Tools -> Parameters ↓	Q TP	Load Runner	WinRunner	Junit
Usability	5	5	5	5
Security	5	4	4	5
Efficiency	5	3	3	4
Accuracy	5	4	4	3
Reliability	5	4	3	3
Robustness	5	5	5	5
Calculation	For Q TP The value of Parameter= $5+5+5+5+5+5/6=30/6=5$ For Load Runner The value of parameter= $5+4+3+4+4+5/6=25/6=4.16$ For WinRunner The value of DDT is= $5+4+3+4+3+5/6=24/6=4$ For Junit The value of DDT is= $5+5+4+3+3+5/6=25/6=4.16$			

(Note: 1- Bad, 2 - Average, 3-Good, 4-Very Good, 5-Excellent)

As per above table it clears that, QTP is excellent tool with quality parameters like Usability, Security, Efficiency, Accuracy, Reliability & Robustness followed by Load Runner and Junit have very Good and WinRunner is Good.

5. Findings

- Recording efficiency of QTP, Load Runner, Junit is Good (3) as compared to WinRunner.
- Junit has Excellent (5) Capability of generation of script followed by QTP has Very good capability (4) whereas Load Runner and Win Runner have very bad capability (1).
- QTP, Load Runner & Win Runner has excellent (5) quality of Access data from external source as compared to Junit.
- QTP and Load Runner has excellent (5) quality to change the data during testing without effecting script as compared to WinRunner and Junit.
- QTP, Load Runner & Win Runner has excellent(5) data driven testing quality
- QTP is excellent (5) data driven testing quality followed by Load Runner (4.6) & WinRunner have Very Good (4.3) and Junit is Very Bad (0.33).
- QTP is Excellent tool (5) regarding testing result report as compared to the other tools like Load Runner (4.6) followed by WinRunner (4.3) and further Junit (2.6).
- QTP is Excellent (5) tool with quality parameters like Usability, Security, Efficiency, Accuracy, Reliability & Robustness followed by Load Runner and Junit have Very Good (4) and WinRunner is Good (3).

6. Conclusion

Automated software testing has become necessity of companies because it saves both time and money. QTP, LoadRunner, WinRunner and Junit all are very good tools for test automation. LoadRunner has easy to use UI and efficient playback. Using one of them can be decided based on the application features and scope of testing. Junit is a java framework for performing unit tests on code. By testing code after very change, programmers can be reassured that changing a small amount of code does not break the larger system. Recording efficiency of QTP is very high as compared to other tools. Junit has Excellent Capability of generation of script as compared to other tools. QTP and Load Runner have excellent quality to change the data during testing without effecting script. QTP, Load Runner & Win Runner has excellent data driven testing quality. QTP is a excellent tool for generating test result as compared the other tools like Load Runner, WinRunner & Junit. WinRunner used to quickly create and run sophisticated automated tests on your application. LoadRunner will be best to use for application with lesser security needs. QTP is best where data security is needed even while testing. WinRunner is best for create reusable test script. Junit is best for testing java based application. Finally it clears that the **Quick Test Pro (QTP)** is the best software testing tool as compared to Load Runner, Win Runner & Junit tools.

7. References

1. J.Emi Retna, Greeshma Varghese, Merlin Soosaiya : A study on Quality parameters of software & Metrics for Evaluation, International Journal of Computer Science and Information Technologies, May - June 2010, ISSN:0976- 6367.
2. Mark Kevitt, BSc : Best Software Test & Quality Assurance Practices in the project Life-cycle, April 2008.
3. Harmeet Kaur, Shahanawaj Ahamad, Gurvinder N. Verma : Identification & Analysis of Parameters for Program Quality Improvement : A reengineering Perspective, Computer Engineering and Intelligent System, ISSN 2222-1719 (Paper) ISSN 2222-2863(Online) Vol.4, No. 5, 2013
4. T. Rajani Devi : Importance of Testing in Software Development Life Cycle, International Journal of Scientific & Engineering Research Volume 3, Issue 5, May-2012, ISSN 2229-5518
5. Shaveta, Sachin kumar, Nitika, Snehlata C.S.E student, Department of Computer Science Engineering,Punjab, International Journal of Computer Science and Information Technologies, Vol. 3 (4) , 2012,4562 – 4567 ISSN:0975-9646
6. Rina & Sanjay Tyagi, DCSA, Kurukshetra University Haryana, India: A Comparative Study of Performance Testing Tools, International Journal of Advanced Research in Computer Science and Software Engineering 5, May 2013, ISSN: 2277 128X
7. Mohd. Ehmer Khan, Farmeena Khan : Importance of Software Testing in Software Development Life Cycle, International Journal of Computer Science Issues, Vol. 11, Issue 2, No 2, March 2014, ISSN (Print): 1694-0814, ISSN (Online): 1694-0784.
8. A. Nirmal Kumar, Dr. B.G. Geetha: A systematic Review of Automated Software Testing Tools, International Journal of Pure and Applied Research in Engineering and Technology, 2013; Volume 2(5): 81-91, ISSN: 2319-507X
9. Introduction to software testing available at <http://www.onestopsoftwaretesting.com>
10. Software Development available at http://en.wikipedia.org/wiki/software_dvelopment