# Software Testing

**Rajat Kumar Bal**

## Introduction

In India itself, Software industry growth has been phenomenal. IT field has enormously grown in the past 50 years. IT industry in India is expected to touch 10,000 crores of which software share is dramatically increasing. Software cost/schedules are grossly inaccurate. Cost overruns of several times, schedule slippage’s by months, or even years are common. Productivity of people has not kept pace with demand added to it; it is due to the shortage of skilled people and the complexity of the software’s. Error rates of released software leave customer dissatisfied…Threatening the very business. So testing is required to make it sure that the software works correctly and can be confidently used or we can say that software testing is necessary to know the defect rate of the developed software.

## What is Software Testing?

Software Testing is the process of executing a program or system with the intent of finding errors. Although crucial to software quality and widely deployed by programmers and testers, software testing still remains an art, due to limited understanding of the principles of software. The difficulty in software testing stems from the complexity of software: we can not completely test a program with moderate complexity. Correctness testing and reliability testing are two major areas of testing. Software testing is a trade-off between budget, time and quality.

Unlike most physical systems, most of the defects in software are design errors, not manufacturing defects. Software does not suffer from corrosion, wear-and-tear -- generally it will not change until upgrades. So once the software is shipped, the design defects -- or bugs -- will be buried in and remain latent until activation. Regardless of the limitations, testing is an integral part in software development. It is broadly deployed in every phase in the software development cycle.

## Goal of Software Testing

- To improve Quality.
- To exercise or evaluate a system by manual or automatic means to verify that system meets the user requirements and check for the systems reliability.
- To identify the correctness, completeness, security and quality of developed computer software.

- To determine the status of the product during and after the build or do component of the process. The role of testing changes as the type of process used to build the product changes.

**Why software testing?**

- To produce quality product.
- To reduce the failure cost and maintenance cost.
- To unearth and correct the defects.
- To detect defects early and to reduce the cost of defect fixing.
- To ensure that product works as user expected it to.
- To avoid user detecting problems.

**Definition**

There are many published definitions of Software testing, however, all of these definitions boil down to specially the same thing: Software Testing is the process of executing software in a controlled manner, in order to answer the question “Does the software behave as specified?”
Software testing is often associated with terms Verification and Validation.

**Verification & Validation**

Verification typically involves reviews and meetings, to evaluate the documents, plans, code, requirements and specifications. Software testing is just one kind of verification.

Validation is the process of checking that what has been specified is what the user actually wanted. Validation typically involves actual testing and takes place after verifications are completed. It can be defined as a “Determination of the correctness of a final program with respect to its requirements”.

**Testing Cycle**

Although testing varies between organizations, there is a cycle to testing:

- Requirements Analysis: Testing should begin in the requirements phase of the software life cycle.
- Design Analysis: During the design phase, testers work with developers in determining what aspects of a design are testable and under what parameter those testers work.
- Test Planning: Test Strategy, Test Plans, Test Bed creation.
- Test Development: Test Procedures, Test Scenarios, Test Cases and Test Scripts to use in testing software.
- Test Execution: Testers execute the software based on the plans and tests and report any errors found to the development team.
- Test Reporting: Once testing is completed, testers generate metrics and make final reports on their test effort and whether or not the software tested is ready for release.

- Retesting The Defects: Not all errors or defects reported must be fixed by a software development team. Some may be caused by errors in configuring the test software to match the development or production environment. Some defects can be handled by a workaround in the production environment. Others might be deferred to future releases of the software, or the deficiency might be accepted by the business user.

**Testing Techniques**

- White Box Testing
- Black Box Testing
- Regression Testing

**White Box Testing:**

Also known as glass box, structural, clear box and open box testing. A software testing technique whereby explicit knowledge of the internal workings of the item being tested are used to select the test data. Unlike black box testing, white box testing uses specific knowledge of programming code to examine outputs. The test is accurate only if the tester knows what the program is supposed to do. He or she can then see if the program diverges from its intended goal.

**Black Box Testing:**

Also known as functional testing. A software testing technique whereby the internal workings of the item being tested are not known by the tester. For example, in a black box test on software design the tester only knows the inputs and what the expected outcomes should be and not how the program arrives at those outputs. The tester does not ever examine the programming code and does not need any further knowledge of the program other than its specifications.

Black Box Testing Techniques

- Boundary Value Analysis: This is a technique used to minimize the test cases. It checks for the corner cases, i.e. One value greater than the maximum and one value less than the minimum.
- Equivalence Partitioning: This is another technique for restricting the test cases. Here test data is divided into valid and invalid classes and both the classes are tested.
- Error Guessing: The use of past experience and an understanding of the weaknesses of human developers.

**Regression Testing:**

Testing done to ensure that the code changes have not had an adverse affect to the other modules or on existing functions. Retesting of a previously tested program following modification to ensure that faults have not been introduced or uncovered as a result of the changes made. Re-running a set of tests that used to work to make sure that changes to the system didn't break anything. It's usually run after each set of maintenance or enhancement changes, but is also very useful for Incremental Integration of a system.

## Levels of Testing

### Unit Testing

- The most 'micro' scale of testing to test particular functions or code modules. Typically done by the programmer and not by testers.
- Unit - smallest testable piece of software.
- Unit testing done to show that the unit does not satisfy the functional specification and/ or its implemented structure does not match the intended design structure.

### Functional Testing

The basic functionality of the application under test is tested under various conditions

- Study SRS
- Identify Unit Functions
- Take each input function
- Identify Equivalence class
- Form Test cases
- Form Test cases for boundary values
- From Test cases for Error Guessing
- Form Unit function Test cases, Cross Reference Matrix
- Find the coverage

### Integration Testing

- This is the phase of software testing in which individual software modules are combined and tested as a group. It follows unit testing and precedes system testing.
- The purpose of Integration testing is to verify functional, performance and reliability requirements placed on major design items

### System Testing

- System testing is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.
- System testing falls within the scope of Black box testing, and as such, should require no knowledge of the inner design of the code or logic.

### 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.

### 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.

**Performance/Load/Stress Testing**

- Performance Testing: In software engineering, performance testing is testing that is performed to determine how fast some aspect of a system performs under a particular workload. Performance testing can serve different purposes. It can demonstrate that the system meets performance criteria. It can compare two systems to find which performs better. Or it can measure what parts of the system or workload cause the system to perform badly. It checks that at what point the system's performance downgrades.

- Load Testing: Load testing is conducted to find out the maximum load that an application can handle.

- Stress Testing: This is done to find out that how the system behaves under abnormal conditions. It is a form of testing that is used to determine the stability of a given system or entity. It involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results.

**Volume Testing**

- Volume tests study the largest tasks the program can deal with. The purpose of Volume Testing is to find weaknesses in the system with respect to its handling of large amounts of data during short time periods. For example, this kind of testing ensures that the system will process data across physical and logical boundaries such as across servers.

**Usability Testing**

- Usability testing is a means for measuring how well people can use some human-made object for its intended purpose, i.e. usability testing measures the usability of the object. Usability testing focuses on a particular object or a small set of objects. During usability testing, the aim is to observe people using the product in as realistic a situation as possible, to discover errors and areas of improvement.
- Usability means that systems are easy and fast to learn, efficient to use, easy to remember, cause no operating errors and offer a high degree of satisfaction for the user. Usability means bringing the usage perspective into focus, the side towards the user.

**Security Testing**

- Application vulnerabilities leave your system open to attacks, Downtime, Data theft, Data corruption and application Defacement. Security within an application or web service is crucial to avoid such vulnerabilities and new threats.
- If your site requires firewalls, encryption, user authentication, financial transactions, or access to databases with sensitive data, you may need to test these and also test your site's overall protection against unauthorized internal or external access.

### User Acceptance Testing

- User acceptance testing (UAT) is one of the final stages of a software project and will often occur before the customer accepts a new system.
- Users of the system will perform these tests which, ideally, developers have derived from the User Requirements Specification, to which the system should conform.

### Testing Best Practices

These are designed to help verify that the product really works and is solid.

- Understand the product architecture before you start testing the product: If you do not understand the architecture and the inner workings of the product you are testing, then you will not be able to anticipate that where it is most error-prone. As a result you could overlook easy opportunities to uncover a large amount of errors in a small amount of time.
- As testers gain experience, they learn that there are some parts of the program that are more prone to errors than others. Similarly they get acquainted with the working styles of the particular group of developers or even an individual developer.
- So just as the Police Detective needs to understand the entire situation before he can solve a murder mystery, you need to understand the entire situation to solve the mystery of “Does this software actually work.”
- Anticipate potential misuses and verify how the software respond to those cases: Don’t think that your job is done once you have verified that the software does what it is expected to do. Users deliberately to use the software in unexpected ways – sometimes because they misunderstand how to use the product (as a new user), sometimes because they see additional usages for the product (as an experienced user), and sometimes because they want to launch the security attacks through the product (as a hacker).
- Record clearly the procedure to reproduce each error found: For each problem that you detected, be sure to record detailed, unambiguous instructions to reproduce the problem. If this is confusing and incomplete, it can create 2 problems.
    1. Developer might not be able to reproduce the error and thus will not be able to fix the error. Even if somehow he manages to reproduce it, so it adds to a lot of waste of time.
    2. You will not be able to verify effectively, whether the developer’s modification corrected the problem. So in that case a passed test case may not necessarily mean that error was corrected.
- Help the team prevent errors: Typically when a tester finds an error, he/she logs that error in the bug tracking system, the responsible and assigned developer tries to reproduce the problem, repair the bug and fix it. Then finally tester has to verify that bug is really fixed and that the changes made do not cause the other related areas behave unexpectedly.
- This is a very traditional and time consuming method. Instead a new method of error prevention can be applied by making the teams aware the type of error occurred and its consequences and the efforts needed to mend this error. In this way we can educate the team that once a kind of error is found then similar kind of error should not reoccur in future and thus this saves the efforts of both: the testing and development team.

### Software Quality Assurance

- The Purpose of software quality assurance is to provide management with appropriate visibility into the processes being used by the software project and of the product being built.
- Software Quality Assurance involves reviewing and auditing the software products and activities to verify that they comply with the applicable procedures and standards and providing the software project and other appropriate managers with the results of these reviews and audits.

### Automation Testing

Automation Testing is no different from a programmer using a coding language to write programs to automate any manual process. Automation testing is very consistent and allows the repetitions of similar test at very little margin cost. The setup and purchase cost of such automation are very high however and maintenance can equally expensive. Automation is relatively inflexible and requires rework in order to adapt to changing requirements.

### Benefits of Automation Testing

1. Fast
2. Reliable
3. Repeatable
4. Programmable
5. Comprehensive
6. Reusable

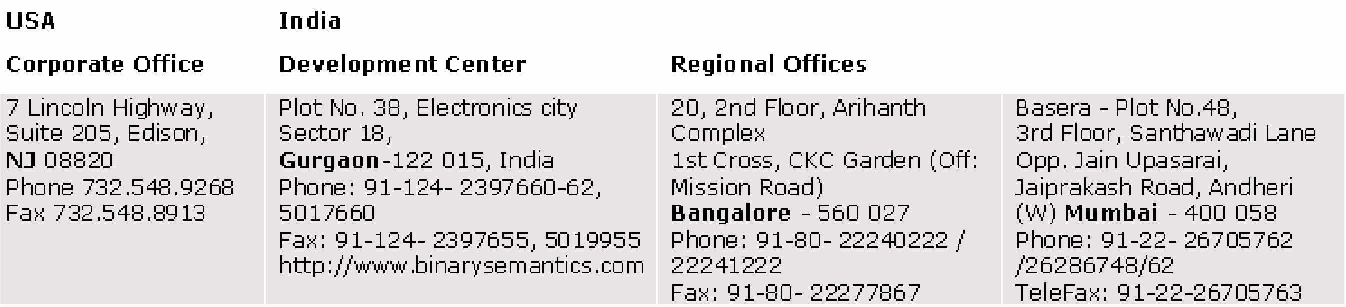
### Tools used for Automation Testing

1. Win Runner
2. Load Runner
3. Test Director
4. Quick Test Professional
5. Silk Test
6. Silk Performer
7. Silk Monitor
8. Silk Pilot
9. Silk Observer
10. Silk Meter
11. Silk Realizer
12. Silk Radar
13. Rational Robot
14. IBM Rational Functional Tester
15. Rational Test Manager
16. Rational ClearQuest
17. QA Center

### Conclusions

- Software testing is an art. Most of the testing methods and practices are not very different. It is nowhere near maturity, although there are many tools and techniques available to use. Good testing also requires a tester's creativity, experience and intuition, together with proper techniques.

- Testing is expensive. Automation is a good way to cut down cost and time.
- Complete testing is infeasible. Complexity is the root of the problem. At some point, software testing has to be stopped and product has to be shipped. The stopping time can be decided by the trade-off of time and budget. Or if the reliability estimate of the software product meets requirement.



| **USA** | **India** | | |
|---|---|---|---|
| **Corporate Office** | **Development Center** | **Regional Offices** | |
| 7 Lincoln Highway,<br>Suite 205, Edison,<br>NJ 08820<br>Phone 732.548.9268<br>Fax 732.548.8913 | Plot No. 38, Electronics city<br>Sector 18,<br>**Gurgaon**-122 015, India<br>Phone: 91-124- 2397660-62, 5017660<br>Fax: 91-124- 2397655, 5019955<br>http://www.binarysemantics.com | 20, 2nd Floor, Arihanth Complex<br>1st Cross, CKC Garden (Off: Mission Road)<br>**Bangalore** - 560 027<br>Phone: 91-80- 22240222 / 22241222<br>Fax: 91-80- 22277867 | Basera - Plot No.48,<br>3rd Floor, Santhawadi Lane<br>Opp. Jain Upasarai,<br>Jaiprakash Road, Andheri (W) **Mumbai** - 400 058<br>Phone: 91-22- 26705762 /26286748/62<br>TeleFax: 91-22-26705763 |