

Software Testing

Introduction

Software is not like a normal physical system. It differs from other systems in the manner in which it fails. Most physical systems fail in a fixed (and reasonably small) set of ways. By contrast, software can fail in many strange ways. Detecting all of the different failure modes for software is generally infeasible. Still, delivering quality products is a very important aspect in the Information Technology industry.

Testing is one instrument that can contribute to the quality enhancement of the software products. Software Testing is the process of executing a program or system with the goal of finding errors. Software Testing is a process of examining the entire software by manual or automated means to find whether it satisfies all the requirements. Software testing is the process used to identify the correctness, completeness and the quality of the developed computer software.

Unlike most physical systems, most of the defects in software are design errors, not manufacturing defects. Software bugs will almost always exist in any software module, not because programmers are careless or irresponsible, but because the complexity of software is generally obdurate, and humans have only limited ability to manage complexity. It is also true that for any complex systems, design defects can never be completely ruled out.

Using the analogy of a medical diagnosis, a successful investigation is one that seeks and discovers a problem, rather than one that reveals nothing and provides a false sense of

well-being. Based on this definition, we establish that a good set of test cases should be one that has a high chance of uncovering previously unknown errors, while a successful test run is one that discovers these errors.

Software Testing is an extremely crucial phase of the software development life cycle (SDLC) and developers today face most demanding customers who expect 100% flawless functioning of the applications in use. In other words, the software product should be free from technical anomalies – ensuring lesser number of complaints, greater customer satisfaction and optimum retention rates. A well-tested application, product or service is, therefore, a necessity to ensure customer satisfaction.

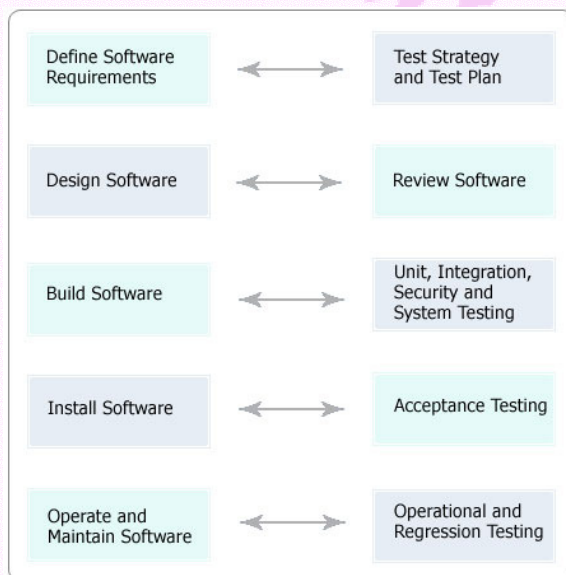
Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Although crucial to software quality and widely deployed by programmers and testers, software testing still remains an art, due to limited understanding 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. Testing is more than just debugging. The purpose of testing can be quality assurance, verification and validation, or reliability estimation.

Software testing has emerged from the shadows in the IT space to claim its rightful place. The Global software Testing market is estimated to be \$13 billion dollars. The market opportunity for the Indian offshore testing

companies is currently \$2 billion, estimated to rise to \$8 billion in 2008. The software-testing arena would require 16,000 – 18,000 professional in the next one year as per the IDC report. Job opportunities are abundant in the testing scene.

India is becoming one of the leading destinations for offshore testing. Most organizations are realizing that software testing is becoming an independent professional discipline. It not only brings objectivity and transparency to defect reporting process, but also improves the core business strategy. Customers are ready to get their software tested by a company that has not played a part in the development process. Moreover it reduces costs by deploying cost-effective models and perceived risk of outsourcing testing is low. Ever increasing competition in the international market has forced organization to adapt to the recent quality practices to match the industry requirements.

Software Testing Process



Software Testing Types

A. Black Box Testing

This testing focuses on the functional requirement of the software. It enables the software engineer to drive set of input condition that will fully exercise all functional requirements for a program. Black Box testing is testing without knowledge of the internal working of the item being tested i.e. when black box testing is applied to software engineering, the tester would know only the legal inputs and what the expected output should be, but not how the program actually arrives at those outputs.

- 1) *Functional Testing* : This Functional testing: This entails a series of tests which perform a feature by feature validation of behavior, using a wide range of normal and erroneous input data. This can involve testing of the product's user interface, APIs, database management, security, installation and networking.
- 2) *Load Testing*: Load testing is a generic term covering Performance Testing and Stress Testing.
- 3) *Stress Testing*: Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements to determine the load under which it fails and how. A graceful degradation under load leading to non-catastrophic failure is the desired result. Often Stress Testing is performed using the same process as Performance Testing but employing a very high level of simulated load.

4) *Performance Testing*: Performance Testing is a problem area because system performance is often poorly specified. Systems performance can be measured using a variety of different criteria, including:

- System Response Time
- External Interface Response Time
- CPU utilization
- Memory utilization

Performance testing can be applied to understand your application or WWW site's scalability, or to benchmark the performance in an environment of third party products such as servers and middleware for potential purchase. This sort of testing is particularly useful to identify performance bottlenecks in high use applications. Performance testing generally involves an automated test suite as this allows easy simulation of a variety of normal, peak, and exceptional load conditions

5) *Ad-hoc Testing*: Ad hoc testing is a term commonly used for the tests carried out without planning software and documentation. The tests are intended to be executed only once, unless a defect is discovered. This is part of the exploratory test, which is the least formal of test methods.

6) *Usability Testing*: Here, users are asked to perform a series of specified business tasks with the application under controlled conditions in order to test the usability goals or requirements of the application under test (AUT).

7) *Smoke Testing*: It is a quick-and-dirty test to check whether the major functions of a piece of software work without bothering with finer details.

8) *Regression Testing*: Similar in scope to a functional test, a regression test allows a consistent, repeatable validation of each new release of a product or Web site. Such testing ensures reported product defects have been

corrected for each new release and that no new quality problems were introduced in the maintenance process. Though regression testing can be performed manually, an automated test suite is often used to reduce the time and resources needed to perform the required testing.

9) *Parallel Testing*: Parallel testing is done by comparing results from two different systems like old versus new or manual versus automated.

10) *Recovery Testing*: Recovery testing is very necessary to check how fast the system is able to recover against any hardware failure, catastrophic problems or any type of system crash.

11) *Acceptance Testing*: The purpose of acceptance testing is to confirm that the system meets its business requirements and to provide confidence that the system works correctly before it is formally handed over to the end users.

12) *Alpha Testing*: It is the software prototype stage when the software is first able to run. It will not have all the intended functionality, but it will have core functions and will be able to accept inputs and generate outputs. An alpha test usually takes place in the developer's offices on a separate system.

13) *Beta Testing*: The software has reached "beta" stage when it is operating with most of its functionality, and is ready for user feedback. Beta tests enable the software to be tested in customer environments, giving users the opportunity to exercise the software, find errors, and correct them before a product is released. For beta test, the software is delivered to customers' sites, along with instructions on what features to exercise in the software. The developer records, tracks, and addresses reported bugs.

14) *Compatibility Testing*: This testing is to ensure compatibility of an application or Web site with different browsers, OSs, and hardware

platforms. Compatibility testing can be performed manually or can be driven by an automated functional or regression test suite. The purpose here is to verify that when the application under test runs in the live environment, its operation does not impact adversely on other systems, and vice versa.

15) Conformance Testing: This testing is done to verify implementation conformance to industry standards. Producing tests for the behavior of an implementation to be sure it provides the portability, interoperability, and/or compatibility a standard defines.

16) Integration Testing: In this testing, modules are combined and tested as a group. Modules are typically code modules, individual applications, client and server applications on a network, etc. Integration Testing follows unit testing and precedes system testing.

17) System Testing: 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.

B. White Box Testing

White box testing (also called as clear box testing, glass box testing, and transparent box testing or structural testing) uses an internal perspective of the system to design test cases based on internal structure. It requires programming skills to identify all paths through the software. The tester chooses test case inputs to exercise paths through the code and determines the appropriate outputs. White box Testing requires knowledge of the internal program structure

Typical white box test design techniques include :

- Control Flow Testing
- Data Flow Testing

- Branch Flow Testing

1) Unit Testing: This testing is used to produce tests for the behavior of components of a product to ensure their correct behavior prior to system integration.

2) Security Testing: Security issues are of the highest concern to many organizations especially those with web-based products and services. Despite this fact, security testing is often the least understood and least well-defined test type. Security testing efforts require domains of expertise beyond traditional software testing. This testing is intended to test whether the features implemented within the AUT provide this required level of protection.

3) Installation Testing: This is to ensure that hardware and software have been correctly installed, all necessary files and connections, have been created, all appropriate data files have been loaded, system default settings have been correctly set, and interfaces to the other hardware systems or components or peripheral devices are all working fine.

4) Mutation Testing: In this type of software testing, the application is tested for the code that was modified after fixing a particular bug/defect.

Roles and Responsibilities in Testing

It is possible for a single person to assume more than one role within any testing process. The number of roles simultaneously existing would depend on the size and scope of the testing activity.

A. Test Manager

This role should be created right from the beginning of the software development project and certainly not later when the requirements gathering phase is over. The Test Manager has the responsibility to administer the organizational aspects of the testing process on a day-to-day basis.

B. Test Team Leader

In small testing projects, the Test manager may himself play this role. However, in larger projects, this role needs to be created separately. The test team Leader is given the responsibility to assign tasks to one or more testers and monitoring their progress against agreed upon plans.

C. Test Analyst

He is responsible for the design and implementation of one or more Test Scripts, which will be used to accomplish the testing of the AUT.

D. Tester

He is primarily responsible for the execution of the Test Scripts. He is also responsible for the recovery of the test environment in the event of failure of the system.

E. Independent Test Observer

He is responsible for providing independent verification that correct procedure is followed during the testing of the AUT

V. Challenges in testing

It is human nature to respond to any challenges thrown upon them. But what do we mean by a challenge? Challenge can be defined as a demanding or stimulating situation.

These challenges can be classified into two broad categories:

A. Analytical Challenges

These challenges are not unique to this profession and most of the professions require some degree of analytical skills. Analytic can be defined as having the ability to analyze or ability to divide into elements and principle. Given the nature of testing, it is very important for testers to have good analytical skills. Like

any other skill, analytical skill can also be improved by practice. This skill can be improved by practicing different puzzles, taking challenges or reading different books to improve your aptitude. Best mechanism of improving this skill is to exercise your brain.

B. Professional Challenges

Every profession has its own challenges. These may be the interaction with the developer, attitude of Management towards testing, identification of tools, their usage and training, interaction with the customers, acceptability and entry-exit criteria for the tests.

VI. Outsourcing Software Testing to India

India is one of the most popular outsourcing locations. Technology companies here possess skills and expertise that are among the finest in the business.

The reasons for choosing India for outsourcing are:

Availability of testers and their use of automated tools

- Indian software testing companies can offer testing services at a fraction of the cost it is in most other parts of the world
- Quality work
- Delivery time

Not all organizations have the expertise or resources to carry out the software testing process. Software testing is essential, but it is definitely not the core activity of most organizations that require it. Outsourcing will enable a company to concentrate on its core activities while software testing experts can handle the work efficiently, ensuring quality results. The company will save time and money on a process that would otherwise be too tedious and exhausting if performed in house.

VII. Indian Software Testing: Problems and Challenges

Indian software testing industry is growing rapidly today. The number of quality certified organizations in India is increasing by leaps and bounds. But all this does not come so easily. The Indian industry has its share of issues and problems that provide a roadblock to its development.

The major problems and challenges faced by the Indian Software Testing Industry are as follows:

- Disproportion in Growth of Outsourcing
- People Management
- Weak Infrastructure
- Rising competition
- Protectionism

Let us see them one by one:

A. Disproportion in Growth of Outsourcing

The growth of India's software testing industry has been tremendous. Outsourcing began in 1974 but caught pace in the 80s. However, everything is not as rosy as it seems. Though the growth rate has been high, it has been quite inconsistent.

The unevenness in outsourcing can be attributed to a lot of factors namely, recession and political turmoil. Also, heavy dependence on the US is another important factor in the disproportionate rise in outsourcing.

Indian companies have clients in more than 40 countries but there is a heavy dependence on the US market. The US market dominates Indian software testing industry partly because it is by far the world's largest software market, comprising approximately half of all software sales throughout the world, and partly because American information technology companies have moved much quickly than

their European counterparts to take advantage of offshore markets. Also the US has more liberal immigration rules for work or residence than most other developed nations. However, India is also more connected with the US market because many Indian businesses have links through their family members or friends who are US residents.

The US dependence might present a limitation to future growth because US share of the world market is slowly reducing. Nevertheless, in the recent years, outsourcing has also increased from other developed European nations.

B. People Management

There is persistent dissatisfaction among senior managers in the software testing industry with the performance of software projects. This takes the form of inordinate delays, time and cost overruns, user discontent and maintenance problems. High burnout is a common problem in most Indian software companies.

In order to overcome these harms, software companies are investing heavily to make improvements in their testing processes and practices. Many of them have realized that durable improvements require significant changes in the way they manage and use people.

In many software companies, the work of managing people is seen as a responsibility of the Human Resources Department. The practices relating to management of people are done in an ad-hoc fashion with little understanding of their impact on the motivation and morale of staff. The reluctance of project leaders or managers to undertake such responsibilities is because they have not been trained to perform functions relating to management of people and also the fact that

they are constantly under the pressure of delivery deadlines.

C. Weak Infrastructure

Infrastructure consists of transport, electricity, land availability, standard of living, government facilities etc. Seen from the eyes of a foreign investor, the perception is that, at present, there are severe limitations in the Indian infrastructure.

Good communication is considered vital for the continued development of the Indian software testing industry. Overall, the data communication infrastructure in India is expensive and in limited supply. For example, government agencies like the department of telecommunications and VSNL are unwilling to give up their vice-like grip on telecommunications and these results in a barrier to the dissemination of Internet.

Another factor that causes negative discernment in the minds of foreign investors is the power grid failures. These failures for several hours at a stretch are a common feature even in the metropolitan cities of India. Also, there are shooting civil wars in progress in several states owing to ethnic hostilities.

There is a silver lining though-foreign investor's feel that in spite of all these problems, India is quietly becoming an international software giant. The Indian government appears to have taken some bold steps towards providing the right combination of incentives along with a relatively low hassle support. Now we have a coordinated, rational strategy for industrial policy. A concern for quality and desire to make effective use of the state of the art technologies is evident.

D. Growing Competition

Currently, India is heading towards becoming one of the biggest software testing countries. But this position may not last for too long. The global reach for software products and services from India is constantly under threat from global competition from other countries that can offer educated manpower at competitive rates. At the moment, China and Russia are the biggest threats to the Indian software testing industry. Also, Philippines and Romania are quite competitive.

E. Protectionism

After the September 11 attack on the US and a spate of terrorist attacks all over the world, India's software testing industry is now discovering new woes-visa restrictions and police raids overseas. This is due to protectionism.

VIII. Future trends in Software Testing

Trends in the industry suggest that software testing in the future will look very different than it does today. The major trends include greater adoption of SOA, web services, wireless and mobile technologies. Each of these trends is further complicated by a more agile approach to software development and an increasing emphasis on the 4Rs; repeatability, reliability, re-use and robustness.

In increasingly automated and "robot" driven engineering environment, the Testing professional will continue to be pivotal in the Software Engineering Lifecycle. Working in multi-disciplinary teams alongside with Subject Matter Experts, Business Analysts, Architects and End Users, they will continue to play the roles of arbitrator, mediator, translator and negotiator between other parties, ensuring that their ideas are not only effectively specified and designed, but that the criteria required for effective quality and testing is captured and articulated into the models and

tools, so that test are accurately generated by the "robots" to verify the design and validate the requirements.

A. Bug Prevention and Early Detection

Because the emphasis is now on the quality of the deliverable (and not on how many bugs you found along the way), prevention practices and detection tools, such as static analyzers, are mainstream.

B. Simulation in Testing

Simulation tools become common, making it easy to "fake" computer environments. Testing for exceptions and error paths now happens early in the development process. After the code has been written, real environments verify that the simulations were accurate.

C. Fewer Testers, Better Testers

Machines now perform much of the mundane work that testers previously did creating tests. Teams require fewer testers, and the testers who remain are more highly trained. Their work is more interesting to them because they are focused on bigger issues in their tests rather than slogging through grunt work.

D. Boundaries blur between Testing and Development

Testers and developers work in tandem to produce testable, high-quality code. Testers help iron out spec issues to make the developer's job easier, and developers create cleaner, more testable code to make the tester's job easier.

E. New Challenges Emerge
The sophisticated and interconnected environment of the computing world guarantees that new problems such as

security testing continue to keep testers running hard. Testers may find these challenges invigorating.

F. Testers Get Respect

Testers are no longer called in at the last moment to "pound on the product". They provide a visible, vital, value-added service throughout the software development process. People realize that testing can be rewarding, interesting, and even fun.

G. Testing Gets Trendy

Software testers start to hold their heads high. And, because breaking stuff is at least as much fun as building it, people begins to rotate between positions in development and testing. Everyone learns more about what makes good code.

IX. Conclusion

Some unexpected troubles in software systems have caused a significant influence on our daily lives. These problems cannot be solved only by debugging, but by conducting various software tests both in and around the system where every possible case is covered. Technical development and system and environmental maintenance should be conducted to create a quality and reliable software, along with the promotion of software test development, fostering its specialists. In fact, due to both the importance and difficulty of software test, new companies and organizations specialized in software test have emerged and their business is steadily growing. These are "industrialization of quality" of software so that it should be emphasized that investment in software test researchers, designers, engineers and managers would not only enhance the industrial competitiveness but also to create a new knowledge industry contributing to the safety and security of the country and the world.