Study of Testing Strategies and availableTools

Shruti N. Pardeshi

Student
Medicaps Institute of Science and Technology, Indore

Abstract- Software testing is a process used for evaluating an attributes or capability of program and make sure that it meets the requirements. Now-a-days testing becomes an very important activity in terms of exposure as well in terms of security, performance and usability. If we consider hardware and software licenses, the testing is too expensive task for user. For the automation of Software testing, several tools are available in the markets that are described further in paper.

Index Terms-Software Testing, Correctness Testing,Box Testing, White Box Testing, Grey Box Testing

I. Introduction

Software testing plays a very significant role in the growth of an enterprise. Over time, the software testing function has become a challenging activity for enterprises due to increasing technological complexities, software sourcing challenges

Typically, software testing is done either internally within the organization, or then is outsourced to software services providers. Testing follows its own evolution cycle. This offered customers a dynamically scalable and economic framework which enabled them to outsource their testing requirements.

II. TESTING

Softwaretesting is activities conducted for finding errors in software. It also verifies and validate whether the program is working correctly with no bugs or not. It analyzes the software for finding bugs. Software testing is not only finds the bugs but also confirms that either the software is working according to the requirement specifications or not.

Software testing will consist of number of steps which is designed to make sure that computer code does what it was designed to do.[1][2]

Software Testing Objectives :

Main goal of testing can be quality assurance, reliability estimation, validation or verification. The other objective of testing includes :

- Testing is the process of executing a program with the intent of finding errors.
- A good test case is one with a high probability of finding an as-yet undiscovered error.
- A successful test is one that discovers an as-yet-undiscovered error.
- The better the software works the more efficiently it can be tested.
- Better the software can be controlled more the testing can be automated and optimized.
- Testing is basically the process to identify the correctness and completeness of the software.[3][4]

Software Testability Checklist

- Operability - the better it works the more efficiently it can be tested.
- Observabilty - what you see is what you test.
- Controllability - the better software can be controlled the more testing can be automated and optimized.
- Decomposability - by controlling the scope of testing, the more quickly problems can be isolated and retested intelligently.
- Simplicity - the less there is to test, the more quickly we can test.
- Stability - the fewer the changes, the fewer the disruptions to testing.
- Understandability - the more information known, the smarter the testing.

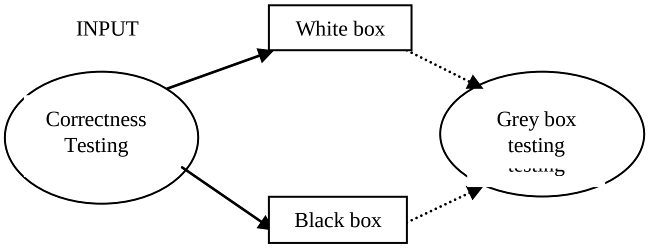
III. TYPES OF TESTING TECHNIQUES

The importance of software testing to software quality cannot be overemphasized. Once source code has been generated, software must be tested to allow errors to be identified And removed before delivery to the customer. While it is not possible to remove every error in a large software package, the software engineer's goal is to remove as many as possible early in the software development cycle. It is important to remember that testing can only find errors. It cannot prove that a program is bug free.

A. *Correctness Testing*

The most essential purpose of testing is correctness which is also the minimum requirement of software. Correctness testing tells the right behavior or execution of the system from the wrong one.

Either the White box or Black box point of view can be taken in testing software as a tester may or may not know the details of the software module which is currently under testing.

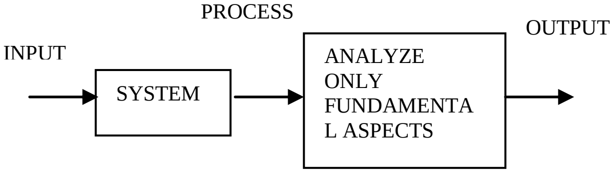


Represent categories of correctness Testing

Black Box Testing

Basically Black box testing is an integral part of Correctness testing but its ideas are not limited to correctness testing only. Correctness testing is a method which is classified by purpose in software testing. Black box testing is based on the analysis of the specifications of a piece of software without reference to its internal working.

The goal is to test how well the component conforms to the published requirement for the component. Black box testing only examines the fundamental aspect of the system. It makes sure that input is properly accepted and output is correctly produced. The black box testing techniques where user involvement is required are user acceptance testing, alpha testing and beta testing.[4]

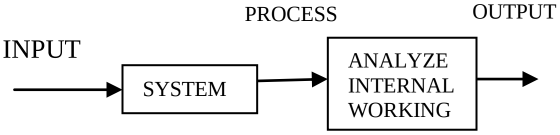


Represent working process of Black Box Testing

White box Testing

White box testing based on an analysis of internal working and structure of a piece of software. White box testing is the process of giving the input to the system and checking how the system processes that input to generate the required output.

It is necessary for a tester to have the full knowledge of the source code. White box testing is applicable at integration, unit and system levels of the software testing process. In white box testing one can be sure that all parts through the test objects are properly executed. [4][6]



Represent working process of White Box Testing

Grey Box Testing

Grey box testing techniques combined the testing methodology of white box and black box. Grey box testing technique is used for testing a piece of software against its specifications but using some knowledge of its internal working as well. [7]

Grey box testing may also include reverse engineering to determine, for instance, boundary values or error messages. Grey box testing is a process which involves testing software while already having some knowledge of its underline code or logic. The understanding of internals of the program in grey box testing is more than black box testing, but less than clear box testing. [2]

B. Functional Testing

This is a type of black box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional Testing of the software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. There are five steps that are involved when testing an application for functionality.

- Step I - The determination of the functionality that the intended application is meant to perform.
- Step II - The creation of test data based on the specifications of the application.
- Step III - The output based on the test data and the specifications of the application.
- Step IV - The writing of Test Scenarios and the execution of test cases.
- Steps V - The comparison of actual and expected results based on the executed test cases. [9]

The categories under functional testing are as follows

1.) *Unit testing –*

Testing of individual software components or modules. Typically done by the programmer and not by testers , as it requires detailed knowledge of the internal program design and code. may require developing test driver modules or test harnesses.

Unit Testingis done at the lowest level. It tests the basic unit of software, which is the smallest testable piece of software, and is often called unit, module, or component interchangeably.

2.) *Integration testing –*

Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

Integration Testingis performed when two or more tested units are combined into a larger structure.The test is often done on both the interfaces between the components and the larger structure being Constructed, if its quality property cannot be assessed from its components.

3.) *System Testing –*

It tends to affirm the end-to-end quality of the entire system. System test is often based on the functional/requirement specification of the system. Non-functional quality attributes, such as Reliability, security, and maintainability, are also checked.[8]

4.) *Acceptance Testing*

It is done when the completed system is handed over from the developers to the customers or users. The purpose of acceptance testing is rather to give confidence that the system is working than to find errors.

a.) Alpha Testing

This test is the first stage of testing and will be performed amongst the teams. Unit testing, integration testing and system testing when combined are known as alpha testing. During this phase Spelling Mistakes, Broken Links, Cloudy Directions will be tested in the application.

b.) Beta testing –

Beta Testing typically done by end-users or others. Final testing before releasing application for commercial purpose. This test is performed after Alpha testing has been successfully performed. In beta testing a sample of the intended audience tests the application. Beta testing is also known as pre-release testing. Beta test versions of software are ideally distributed to a wide audience on the Web, partly to give the program a "real-world" test and partly to provide a preview of the next release.

C. Non-Functional Testing

Non-functional testing of Software involves testing the Software from the requirements which are non-functional in nature related but important as well, such as performance, security, user interface etc. Some of the important and commonly used non-functional testing types are mentioned as follows.

1.) Performance Testing

Term often used interchangeably with 'stress' and 'load' testing. To check whether system meets performance requirements. Used different performance and load tools to do this.

'Performance Testing' involve all the phases as the mainstream testing life cycle as an independent discipline which involve strategy such as plan, design, execution, analysis and reporting. This testing is conducted to evaluate the compliance of a system or component with specified performance requirement. [4]
Evaluation of a performance of any software system includes resource usage, throughput and stimulus response time.

Goals of performance testing:

- Observing the system resources which are under various loads.
- Measuring response time of end to end transactions.
- Measurement of the delay of network between client and server. [5]

Mistakes which happen during performance testing:

- Ignoring of errors in input.
- Analysis is too complex.
- Erroneous analysis.
- Level of details is inappropriate.
- Ignore significant factors.
- Incorrect Performance matrix.
- Important parameter is overlooked.
- Approach is not systematic. [5]

Typically to debug applications, developers would execute their applications using different execution stream. Which are completely exercised the applications in an attempt to find errors. Performance testing is categorized in two types shown below.

a.) Stress Testing

System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.

Stress testing is a testing, which is 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. Stress testing also determines the behavior of the system as user base increases. [5][4]

b.) Load Testing

It is an industry term for the effort of performance testing. The main feature of the load testing is to determine whether the given system is able to handle the anticipated no. of users or not. This can be done by making the virtual user to exhibit as real user so that it will be easy to perform load testing.The main objective of load testing is to check whether the system can perform well for specified user or not.
Two ways for implementing load testing are

1. Manual Testing: It is not a very practical option as it is very iterative in nature and it involves.It will Measure response time and Compare results
2. Automated Testing: As compared to manual load testing the automated load testing tools provide more efficient and cost effective solutions. Because with automated load testing, tools test can easily be rerun any number of times and decreases the chances of human error during testing. [5]

2.) Reliability Testing

Reliability Testing is very important, as it discover all the failures of a system and removes them before the
system is deployed. Reliability testing is related to many aspects of software in which testing process is included;
this testing process is an effective sampling method to measure software reliability.[4][7]

Robustness testing and stress testing are the variances of reliability testing. By Robustness we mean how software component works under stressful environmental conditions. Robustness testing only watches the robustness problem such as machine crashes, abnormal terminations etc. Robustness testing is very portable and scalable. [7]

3.) Security Testing

Security Testing makes sure that only the authorized person can access the program and only the authorized personnel can access the functions available to their security level. Security testing is very helpful for the tester for finding and fixing of problems. It ensures that the system will run for a long time without any major problem. It also ensures that the systems used by any organization are secured from any unauthorized attack.[3]

4.) Portability Testing

Portability testing includes the testing of Software with intend that it should be re-useable and can be moved from another Software as well. Following are the strategies that can be used for Portability testing.
Transferred installed Software from one computer to another.

Following are some pre-conditions for Portability testing:

- Software should be designed and coded, keeping in mind Portability Requirements.
- Unit testing has been performed on the associated components.
- Integration testing has been performed.
- Test environment has been established. [6]

IV. TESTING TOOLS

As the software industry grows. It becomes more competitive and advanced for businesses to produce such good quality software. With this reliability and deadlines should also met. Manual testing takes too long time and it can waste the lot of time.With the aid of testing tools this can increase efficiency and get the deadlines met.Testing tool is a form of automated testing. It is basically program to do various testing tasks.

Now-a-days testing is done with the help of various testing tools. The testing tools are categorized according to its software specification. After studying all the testing tools I got all types of software, languages and tools through which the software can be developed.[9]

This categorization of testing tools according to its software specification is shown in the table below:

Sr. No	Testing Tools	Specification
1	Bound Checker	C,C++,.NET,ASP,ASP.NET
2	AQTime	VB,VC++,C++,GCC,Visual
3	DevAdvantages	C#,.NET
4	GJ-Coverage	J2ME,J2SE,J2EE,HTML
5	ABBOT	JAVA GUI's
6	Cactus	Server-Side java code
7	LDRA Testbed	ADA,ALGOL,C/C++,COBOL, FORTRON, PASCAL,PL/M,PL/I
8	AppLoader	CITRIX,JAVA,.NET,ORACLE,SAP, EPIC ECLIPSYS,MCKESSEN,MEDITECH
9	SQA Suite	POWER BUILDER, DEVELOPER/2000
10	SQA TestFoundation	PEOPLESOFT
11	TCMON	ADA83
12	TestWorks	F77,ADA,C,C++
13	CSE-HTML Validator	HTML,XHTML, CSS, LINK, Spelling & Accessibility checker
14	LinkScan	Dead link Detector
15	Datatect	RDBMS including ALE,SYBASE, SQLSERVER,INFORMIX
16	SQL DB Validator	Performs database and data cube verification and validation and works with SQL
17	UtPLSQL	Open source and works on ORACLE PL/SQL

Table: Description of various Testing tools

V. CONCLUSION

In this paper I discussed about the various strategies used for testing along with its tools available for testing. Testing is very vast and hot topic in the field of research .Here I described the necessity of Testing with tools.

In future I will include more number of testing techniques in support of different fields where testing is used.

REFERENCES

[1] Introduction to software testing available at /www.onestoptetsing.com/introduction/

[2] Software testing techniques available at http://pesona.mmu.edu.my/~wruslan/SE3/Readings/GB1/pdf/ch14-GB1

[3] Different forms of software testing techniques for finding errors. paper by Mohd.Ehmer Khan

[4]Software testing glossary available at http://www.aptest.com/glossary.html#performance testing

[5] Software Testing by Cognizant Technology Solution.

[6] Software testing tutorials by tutorials.com

[7] Paper on Software testing by Jiantao Pan

[8] Software Testing Techniques –Technology maturation and research strategy Paper by Lu Luo.

[9] Research on testing tools available at http://research.cs.queensu.ca/~shepard/testing.dir

Shruti N Pardeshi was born in Jalgaon in 1988.She received her B.E. degree in Computer from North Maharashtra University,Jalgaon in 2009 and she is pursuing her M.Tech in Information Technology from RGPV University,Bhopal.
Email address :shruti1.pardeshi@gmail.com