International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 13
October 2017

Comparison And Analysis Of Testing Types, Processes, Levels And Methods For Software Testing

Myint Myint Moe
Faculty of Information Science, University of Computer Studies, Yangon, Myanmar
myintmyintmoe.ucsy.1971@gmail.com

ABSTRACT: *Software testing performed to verify that the completed software package functions according to the expectations defined by the requirements/ specifications. Testing allows developers to deliver software that meets expectations, prevents unexpected results and improves the long term maintenance of the application. It is to assess or evaluate the capabilities or attributes of software program's ability to adequately meet the applicable standards and customer needs the goals of testing is to find bugs and to equalize the quality of the product. Software testing is some activities aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Although sacrosanct to software quality and widely deployed by programmers and testers, software testing still remains a carving, due to limited understanding of the principles of software. The difficulty in software testing stems from the complexity of software: we cannot completely test a program with moderate complexity. Testing is more than just debugging. The intent of testing can be quality assurance, verification and validation, or reliability estimation. Software testing is to detect software failures so that defects may be discovered and corrected. Software testing began as an integrated process and has grown into an exclusive discipline with multiple specializations and dedicated array of tools. Software testing is a process of executing a program or application with the aim of finding the software bugs.*

Key words: software testing, testing types, processes, levels, methods.

1. INTRODUCTION

The major purposes of Software testing are as follows: Searching defects which may get created by the programmer while developing the software; Getting confidence in and providing information about the level of quality; To prevent defects; To make sure that the end result meets the business and user requirements; To ensure that it mediates Business Requirement Specification (BRS) and System Requirement Specifications(SRS) [9]. To get the confidence of the customers by providing them a quality product. A fundamental problem of software testing is that testing under all combinations of inputs and preconditions are not feasible, even with a simple product. This means that the number of defects in a software product can be very large and defects that occur infrequently are difficult to find in testing. More manifestly, non-functional dimensions of quality : usability, scalability, performance, compatibility, reliabilitycan be highly subjective; Software developers can't test everything, but they can use combinatorial test design to identify the minimum number of tests needed to get the coverage they want. Combinatorial test design enables users to get greater test coverage with fewer tests. Whether they are looking for speed or test depth, they can use combinatorial test design methods to build structured variation into their test cases. This system is expected to perform correctly using a given set of test cases that reflect the systems expected use and to behave after testing the functionality of an application. Actual is what the System behaved after testing the functionality of an application [1]. Software Testing has different goals and objectives. Software testing helps in finalizing the software application or product against business and user requirements. It is very important to have good test coverage in order to test the software application completely and make it sure that it's performing well and as per the specifications. Software testing makes sure that the testing is being done properly and hence the system is ready for use. The testing has been done to cover the various areas like functionality of the application, equitable of the application with the operating system, hardware and different types of browsers, performance testing to test the performance of the application and load testing to make sure that the system is reliable. It also determines that the application can be deployed easily to the machine and without any resistance [7]. It can also be stated as the process of validating and verifying that a software program or application or product: Meets the business and technical requirements that coached it's design and development; Works as expected; Can be implemented with the same characteristic. Software testing is intended to verify the error, bug, failure, issue or differences in software for a particular input and its output which appraises the characteristics of software product. Testing is a process intended to build confidence in the software. [6] This paper is organized in six sections: In section 1.Introduction; 2. Literature review; 3. Testing types; 4. Testing process; 5. Testing levels; 6.Testing methods; 7.Conclusion.

2. LITERATURE REVIEW

Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of searching software bugs. Software testing can also be settled as the process of validating and verifying that a software program / application/ product.[9] Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the software development methodology settled. Most of the test execution occurs after the requirements have been defined and the coding process has been completed. Software testing necessary because we all make mistakes. Some of those mistakes are unimportant, but some of them are expensive or dangerous. We need to check everything and anything we produce because things can always go wrong – humans make mistakes all the time. Exclusively, we should get someone else to check our work because another person is more likely to spot the flaws. There are several reasons which clearly tell us as why Software Testing is important and what are the major things that we should consider while testing of any product or application [4]. Software testing is very important because of the following reasons: Software testing is really required to point out the defects and errors that were made during the development phases. It’s essential since it makes sure of the Customer’s reliability and their satisfaction in the application. It is very important to ensure the Quality of the product. Quality product delivered to the customers helps in gaining their confidence [5]. Testing is necessary in order to provide the facilities to the customers like the delivery of high quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results. Testing is required for an effective performance of software application or product. It’s important to ensure that the application should not result into any failures because it can be very expensive in the future or in the later stages of the development. It’s required to stay in the business. [7]

3. TESTING TYPES

There are many software testing types. Latest updated software testing types are black box testing, white box testing, unit testing, incremental integration testing, integration testing, functional testing, system testing, end-to-end testing, sanity testing, regression testing, acceptance testing, load testing, stress testing, performance testing, usability testing, install/uninstall testing, recovery testing, security testing, compatibility testing, comparison testing, alpha testing and beta testing.

A. Black box testing: Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.

B. White box testing: This testing is based on knowledge of the internal logic of an application’s code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

C. 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. It may require developing test driver modules or test harnesses.

D. Incremental integration testing: Bottom up approach for testing i.e. continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately; Done by programmers or by testers.

E. 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.

F. Functional testing: This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black-box type testing geared to functional requirements of an application.

G. System testing: Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.

H. End-to-end testing: Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

I. Sanity testing: Testing to determine if a new software version is performing well enough to accept it for a major testing effort. If application is crashing for initial use then system is not stable enough for further testing and build or application is assigned to fix.

J. Regression testing: Testing the application as a whole for the modification in any module or functionality. Difficult to cover all the system in regression testing so typically automation tools are used for these testing types.

K. Acceptance testing: Normally this type of testing is done to verify if system meets the customer specified requirements. User or customer do this testing to determine whether to accept application.

L. Load testing: It is a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system’s response time degrades or fails.

M. 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.

N. 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.

O. Usability testing: User-friendliness check. Application flow is tested, Can new user understand the application easily, Proper help documented whenever user stuck at any point. Basically system navigation is checked in this testing.

P. Install/uninstall testing: Tested for full, partial, or upgrade install/uninstall processes on different operating systems under different hardware, software environment.

Q. Recovery testing: Testing how well a system recovers from crashes, hardware failures, or other catastrophic problems.

R. Security testing: Can system be penetrated by any hacking way. Testing how well the system protects against unauthorized internal or external access. Checked if system, database is safe from external attacks.

S. Compatibility testing: Testing how well software performs in a particular hardware/ software/ operating system/ network environment and different combination s of above.

T. Comparison testing: Comparison of product strengths and weaknesses with previous versions or other similar products.

U. Alpha testing: In house virtual user environment can be created for this type of testing. Testing is done at the end of development. Still minor design changes may be made as a result of such testing.

V. Beta testing: Testing typically done by end-users or others. Final testing before releasing application for commercial purpose. [8]

4. FUNDAMENTAL TEST PROCESS IN SOFTWARE TESTING

Testing is a process rather than a single activity. This process starts from test planning then designing test cases, preparing for execution and evaluating status till the test closure. So, we can divide the activities within the fundamental test process into the following basic steps: Planning and Control; Analysis and Design; Implementation and Execution; Evaluating exit criteria and Reporting; Test Closure activities.

A. Planning and Control

Test planning has following major tasks: To determine the scope and risks and identify the objectives of testing; to determine the test approach; to implement the test policy and/or the test strategy. (Test strategy is an outline that describes the testing portion of the software development cycle. It is created to inform PM, testers and developers about some key issues of the testing process. This includes the testing objectives, method of testing, total time and resources required for the project and the testing environments.). To determine the required test resources like people, test environments, PCs, etc. To schedule test analysis and design tasks, test implementation, execution and evaluation. To determine the exit criteria we need to set criteria such as coverage criteria. (Coverage criteria are the percentage of statements in the software that must be executed during testing. This will help us track whether we are completing test activities correctly. They will show us which tasks and checks we must complete for a particular level of testing before we can say that testing is finished.) Test control has the following major tasks: To measure and analyze the results of reviews and testing. To monitor and document progress, test coverage and exit criteria. To provide information on testing. To initiate corrective actions. To make decisions.

B. Analysis and Design

Test analysis and Test Design has the following major tasks: To review the test basis**.** (The test basis is the information we need in order to start the test analysis and create our own test cases. Basically it’s a documentation on which test cases are based, such as requirements, design specifications, product risk analysis, architecture and interfaces. We can use the test basis documents to understand what the system should do once built); to identify test conditions; to design the tests; to evaluate testability of the requirements and system.To design the test environment set-up and identify and required infrastructure and tools.

C. Implementation and Execution

During test implementation and execution, we take the test conditions into test cases and procedures and other test ware such as scripts for automation, the test environment and any other test infrastructure. (A test case is a set of conditions under which a tester will determine whether an application is working correctly or not.) Test ware is a term for all utilities that serve in combination for software testing like scripts, the test environment and any other test infrastructure for later reuse.)Test implementation has the following major task: To develop and prioritize our test cases by using techniques and create test data for those tests. (In order to test a software application you need to enter some data for testing most of the features. Any such specifically identified data which is used in tests is known as test data.)We also write some instructions for carrying out the tests which is known as test procedures.We may also need to automate some tests using test harness and automated tests scripts. (A test harness is a collection of software and test data for testing a program unit by running it under different conditions and monitoring its behavior and outputs). To create test suites from the test cases for efficient test execution.(Test suite is a collection of test cases that are used to test a software program to show that it has some specified set of behaviors. A test suite often contains detailed instructions and information for each collection of test cases on the system configuration to be used during testing. Test suites are used to group similar test cases together). To implement and verify the environment.Test execution has the following major task: To execute test suites and individual test cases following the test procedures; To re-execute the tests that previously failed in order to confirm a fix. This is known as confirmation testing or re-testing; to log the outcome of the test execution and record the identities and versions of the software under tests. The test log is used for the audit trial. (A test log is nothing but, what are the test cases that we executed, in what order we executed, who executed that test cases and what is the status of the test case (pass/fail). These descriptions are documented and called as test log).To compare actual results with expected results.Where there are differences between actual and expected results, it report discrepancies as incidents.

D. Evaluating Exit criteria and Reporting

Based on the risk assessment of the project we will set the criteria for each test level against which we will measure the “enough testing”. These criteria vary from project to project and are known as exit criteria.Exit criteria come into picture, when:Maximum test cases are executed with certain pass percentage; Bugrate falls below certain level; When achieved the deadlines.Evaluating exit criteria has the following major tasks: To check the test logs against the exit criteria specified in test planning; To assess if more test are needed or if the exit criteria specified should be changed; To write a test summary report for stakeholders.

E. Test Closure activities

Test closure activities are done when software is delivered. The testing can be closed for the other reasons also like:When all the information has been gathered which are needed for the testing; When a project is cancelled; When some target is achieved; When a maintenance release or update is done.Test closure activities have the following major tasks:To check which planned deliverables are actually delivered and to ensure that all incident reports have been resolved.To finalize and archive testware such as scripts, test environments, etc. for later reuse.To handover the testware to the maintenance organization. They will give support to the software.To evaluate how the testing went and learn lessons for future releases and projects. [7]

5. SOFTWARE TESTING LEVELS

Testing levels are basically to identify missing areas and prevent overlap and repetition between the development life cycle phases. In software development life cycle models there are defined phases like requirement gathering and analysis, design, coding or implementation, testing and deployment. Each phase goes through the testing. Hence there are various levels of testing. The various levels of testing are:Big bang integration testing; Top down; Bottom up; Functional incremental. Latest updated software testing levels are component integration testing, system integration testing, system testing, alpha testing and beta testing.

A. **Component integration testing**: Both the modules and components are integrated then the testing done is called as Component integration testing. This testing is basically done to ensure that the code should not break after integrating the two modules.

B. **System integration testing**: System integration testing (SIT) is a testing where testers basically test that in the same environment all the related systems should maintain data integrity and can operate in coordination with other systems.

C. **System testing**: In system testing the testers basically test the compatibility of the application with the system.

D. **Acceptance testing**: Acceptance testing are basically done to ensure that the requirements of the specification are met.

E. **Alpha testing**: Alpha testing is done at the developer’s site. It is done at the end of the development process.

F. **Beta testing:** Beta testing is done at the customers site. It is done just before the launch of the product. [9]

6. SOFTWARE TESTING METHODS

There are different methods that can be used for software testing. This popular method briefly describes the methods available.

A. Black-Box Testing

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.This system is intended to verify the error, bug, failure, issue or differences in software for a particular input and its output which appraises the characteristics of software product. Testing is a process intended to build confidence in the software. This system is intended to verify the error, bug, failure, issue or differences in software for a particular input and its output which appraises the characteristics of software product. Testing is a process intended to build confidence in the software.
Advantages of Black-Box Testing: Well suited and efficient for large code segments. Code access is not required. Clearly separates user's perspective from the developer's perspective

through visibly defined roles. Large numbers of moderately skilled testers can test the application with no knowledge of implementation, programming language, or operating systems.

Disadvantages of Black-Box Testing: Limited coverage, since only a selected number of test scenarios is actually performed. Inefficient testing, due to the fact that the tester only has limited knowledge about an application. Blind coverage, since the tester cannot target specific code segments or error-prone areas.The test cases are difficult to design.

B. White-Box Testing

White-box testing is the detailed investigation of internal logic and structure of the code.White-box testing is also called glass testing or open-box testing. In order to perform white-box testing on an application, a tester needs to know the internal workings of the code. The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

Advantages of White Box Testing: As the tester has knowledge of the source code, it becomes very easy to find out which type of data can help in testing the application effectively. It helps in optimizing the code. Extra lines of code can be removed which can bring in hidden defects. Due to the tester's knowledge about the code, maximum coverage is attained during test scenario writing.

Disadvantages of White Box Testing: Due to the fact that a skilled tester is needed to perform white-box testing, the costs are increased. Sometimes it is impossible to look into every nook and corner to find out hidden errors that may create problems, as many paths will go untested. It is difficult to maintain white-box testing, as it requires specialized tools like code analyzers and debugging tools.

C. Grey-Box Testing

Grey-box testing is a technique to test the application with having a limited knowledge of the internal workings of an application. In software testing, the phrase *the more you know, the better* carries a lot of weight while testing an application. Mastering the domain of a system always gives the tester an edge over someone with limited domain knowledge. Unlike black-box testing, where the tester only tests the application's user interface; in grey-box testing, the tester has access to design documents and the database. Having this knowledge, a tester can prepare better test data and test scenarios while making a test plan.

Advantages of Gray-Box Testing: Offers combined benefits of black-box and white-box testing wherever possible. Grey box testers don't rely on the source code; instead they rely on interface definition and functional specifications. Based on the limited information available, a grey-box tester can design excellent test scenarios especially around communication protocols and data type handling. The test is done from the point of view of the user and not the designer.

Disadvantages of Gray-Box Testing: Since the access to source code is not available, the ability to go over the code and test coverage is limited. The tests can be redundant if the software designer has already run a test case. Testing every possible input stream is unrealistic because it would take an unreasonable amount of time; therefore, many program paths will go untested. [8]

D. A Comparison of Testing Methods

The points of differentiate black-box testing, grey-box testing and white-box testing.

Black-Box Testing	Grey-Box Testing	White-Box Testing
The internal workings of an application need not be known.	The tester has limited knowledge of the internal workings of the application.	Tester has full knowledgeof the internal workings of the application.
Also known as closed-box testing, data-driven testing, or functional testing.	Also known as translucenttesting, as the tester has limited knowledge of the insides of the application.	Also known as clear-boxtesting, structural testing,or code-based testing.
Performed by end-usersand also by testers and developers.	Performed by end-users and also by testers and developers.	Normally done by testers and developers.
Testing is based on external expectations - Internal behavior of theapplication is unknown.	Testing is done on the basis of high-level database diagrams and data flow diagrams.	Internal workings are fullyknown and the tester candesign test dataaccordingly.
It is exhaustive and the least time-consuming.	Partly time-consuming and exhaustive.	The most exhaustive and time-consuming type of testing.

Not suited for algorithm testing.	Not suited for algorithm testing.	Suited for algorithm testing.
This can only be done by trial-and-error method.	Data domains and internal boundaries can be tested, if known.	Data domains and internal boundaries can be better tested

7. CONCLUSION

Software testing research is the driving element of development and application. In this area of new and higher demand of software testing, it is important to constantly summarize new achievements, fresh hotspots and propose different ideas in order to promote the study on software testing system engineering, to facilitate the rapiddevelopment on software testing field and industry. Software testing is expected to perform correctly using a given set of test cases that reflect the systems expected use and to behave after testing the functionality of an application. Actual is what the System behaved after testing the functionality of an application. Testing is intended to verify the error, bug, failure, issue or differences in software for a particular input and its output which appraises the characteristics of software product. Testing is a process intended to build confidence in the software.

REFERENCE

[1]Ian Somerville: Software Engineering, Eighth Edition (ISBN 13: 978-0-321-31379-9, ISBN 10: 0-321-31379-8).

[2] Kaner, Cem; Falk, Jack; Nguyen, Hung Quoc (1999). Testing ComputerSoftware, 2nd Ed. New York, et al: John Wiley and Sons, Inc. p. 480. ISBN 0-471-35846-0.

[3] Jiantao Pan, ‘Software Testing’, Carnegie Mellon University , 18-849b Dependable Embedded Systems , Spring 1999 , jpan@cmu.edu.

[4][Hetzel88] Hetzel, William C., The Complete Guide to Software Testing, 2nd ed. Publication info: Wellesley, Mass: QED Information Sciences, 1988. ISBN: 0894352423.Physical description: ix.

[5] Gaurav Shankar, Quality Engineer at Sunquest Information Systems, Nov 7, 2014.

[6]"Proceedings from the 5th International Conference on Software Testing and Validation (ICST). Software Competence Center Hagenberg. "Test Design: Lessons Learned and Practical Implications."

[7]http://istqbeexamcertification.com

[8] http://www.cigniti.com

[9] http://tutorialspoint.com

[10] http:// testing mentor.com

[11] www.rightHand Technologies

[12] www.quora.com