

Software Testing Techniques and Strategies

Isha, Sunita Sangwan

(Department of computer science) SBMNE College, Rohtak INDIA

ABSTRACT

Software testing provides a means to reduce errors, cut maintenance and overall software costs. Numerous software development and testing methodologies, tools, and techniques have emerged over the last few decades promising to enhance software quality. This paper describes Software testing, need for software testing, Software testing goals and principles. Further it describe about different Software testing techniques and different software testing strategies

Keywords – Debugging, Software Testing Goals, Software Testing Life Cycle, Software Testing strategies, Software Testing Techniques.

I. INTRODUCTION

Software testing is a most often used technique for verifying and validating the quality of software. Software testing is the procedure of executing a program or system with the intent of finding faults. Testing is a process of confirming that the product that has been manufactured by programmers is a quality product and to assure that the manufactured product is working according to the specification and satisfying the customer needs.

Software testing is a significant activity of the Software development life cycle (SDLC). It helps in developing the confidence of a developer that a program does what it is intended to do so. In other words, we can say it's a process of executing a program with intends to find errors. In the language of Verification and Validation (V&V), black box testing is often used for validation (i.e. are we building the right software?) and white box testing is often used for verification (i.e. are we building the software right?) . This study emphasizes the need to investigate various testing techniques in software testing field. The definition of testing according to the ANSI/IEEE 1059 standard is that testing is the process of analyse a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

The purpose of testing is verification, validation and error detection in order to find problems – and the purpose of finding those problems is to get them fixed.

1.1. Software testing Goals

- A good test case is one that has a probability of finding an as yet undiscovered error.
- A good test is not redundant.
- A successful test is one that uncovers a yet undiscovered error.

- A good test should neither be too simple nor too complex.
- To check if the system does what it is expected to do.
- To check if the system is “Fit for purpose”.
- To check if the system meets the requirements and be executed successfully in the Intended environment.

1.2. Testing principles

Principle is the rule or method in action that has to be followed. Different testing principles are as follows:

- 1) est a program to try to make it fail
Testing is the process of executing a program with the intent of finding errors. We should expose failures to make testing process more effective.
- 2) Start testing early This helps in fixing enormous errors in early stages of development, reduces the rework of finding the errors in the initial stages.
- 3) Testing is context dependant Testing should be appropriate and different for different points of time.
- 4) Define Test Plan Test Plan usually describes test scope, test objectives, test strategy, test environment, deliverables of the test, risks and mitigation, schedule, levels of testing to be applied, methods, techniques and tools to be used. Test plan should efficiently meet the needs of an organization and clients as well.
- 5) Design Effective Test cases Test case must be specified in a way that is measurable so that testing results are unambiguous.
- 6) Test for valid as well as invalid Conditions In addition to valid inputs, we should also test system for invalid and unexpected inputs/conditions

- 7) Testing must be done by different persons Different purpose addressed at different level of testing so different person should perform testing differently using different testing techniques at different level.
- 8) End of Testing Testing has to be stopped somewhere. The testing can be stopped when risk is under some limit or if there is limitation.

input is properly accepted and output is correctly produced.

Some Different types of Black box testing techniques are as follows:-

- 1) Equivalent Partitioning
- 2) Boundary value Analysis
- 3) Cause-Effect Graphing Techniques
- 4) Comparison Testing

II. SOFTWARE TESTING LIFE CYCLE-PHASES

1. Requirements study

- Testing Cycle starts with the study of client's requirements.
- Understanding of the requirements is very essential for testing the product.

2. Test Case Design and Development

- Component Identification
- Test Specification Design
- Test Specification Review

3. Test Execution

- Code Review
- Test execution and evaluation
- Performance and simulation

4. Evaluating exit criteria

- Test summary report
- Project Documentation

5. Test Closure

- Analysis done on the reports and improving the application's performance by implementing new technology and additional features.

Advantages of Black box testing:-

- 1) The number of test cases are reduced to achieve reasonable testing
- 2) The test cases can show presence or absence of classes of errors.
- 3) Black box tester has no "bond" with the code.
- 4) Programmer and tester both are independent of each other.
- 5) More effective on larger units of code than clear box testing.

Disadvantages of Black box testing:-

- 1) Test cases are hard to design without clear specifications.
- 2) Only small numbers of possible input can actually be tested.
- 3) Some parts of the back end are not tested at all.
- 4) Chances of having unidentified paths during this testing
- 5) Chances of having repetition of tests that are already done by programmer

2) White box testing

White box testing is highly effective in detecting and resolving problems, because bugs can often be found before they cause trouble. 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. White box testing is also called white box analysis, clear box testing or clear box analysis. White box testing is applicable at integration, unit and system levels of the software testing process. White box testing is considered as a security testing method that can be used to validate whether code implementation follows intended design, to validate implemented security functionality, and to uncover exploitable vulnerabilities.

Some Different types of white box testing techniques are as follows:-

- 1) Basis Path Testing
- 2) Loop Testing
- 3) Control Structure Testing

Advantages of white box testing:-

- 1) All independent paths in a module will be exercised at least once.
- 2) All logical decisions will be exercised.

III. TESTING METHODOLOGIES AND TYPES

Here, I have considered the two testing methodology and types that is mention above:

1. Black box testing
2. White box testing

1) Black box testing

Black box testing is testing software based on output requirements and without any knowledge of the internal structure or coding in the program. Basically Black box testing is an integral part of „Correctness testing“ but its ideas are not limited to correctness testing only. The goal is to test how well the component conforms to the published requirement for the component. Black box testing have little or no regard to the internal logical structure of the system, it only examines the fundamental aspect of the system. It makes sure that

- 3) All loops at their boundaries will be executed.
- 4) Internal data structures will be exercised to maintain their validity.
- 5) Errors in hidden codes are revealed.
- 6) Approximate the partitioning done by execution equivalence.

Disadvantages of white box testing:-

- 1) Missed out the cases omitted in the code.
- 2) As knowledge of code and internal structure is a prerequisite, a skilled tester is needed to carry out this type of testing, which increases the cost.
- 3) And it is nearly impossible to look into every bit of code to find out hidden errors, which may create problems, resulting in failure of the application.

IV. SOFTWARE TESTING STRATEGIES

A strategy for software Testing integrates software test case design methods into a well planned Series of steps that result in successful Construction of software that result in successful construction of software. Software testing Strategies gives the road map for testing. A software testing Strategy should be flexible enough to promote a customized testing approach at same time it must be right enough. Strategy is generally developed by project managers, software engineer and testing specialist. There are four different software testing strategies.

- 1) Unit testing
- 2) Integration testing
- 3) Acceptance/Validation testing
- 4) System testing

3.1. Unit testing

Unit is the smallest module i.e. smallest collection of lines of code which can be tested. Unit testing is just one of the levels of testing which go together to make the big picture of testing a system. IT complements integration and system level testing. It should also complement code reviews and walkthroughs. Unit testing is generally seen as a white box test class. That is it is biased to looking at and evaluating the code as implemented. Rather than evaluating conformance to some set of requirements.

Benefits of Unit Testing:-

- 1) Unit level testing is very cost effective.
- 2) It provides a much greater reliability improvement for resources expanded than system level testing. In Particular, it tends to reveal bugs which are otherwise insidious and are often catastrophic like the strange system crashes that occur in the field when something unusual happens.
- 3) Be able to test parts of a project without waiting for the other parts to be available,

- 4) Be able to detect and remove defects at a much less cost compared to other later stages of testing,

3.2. Integration testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design.

Different Integration testing Strategies are discussed below:-

- 1) Top down Integration testing
- 2) Bottom up Integration testing

Top down Integration Top-down integration testing is an incremental approach to construct program structure. Modules are integrated by moving downward through the structure, beginning with the main control module. Modules subordinate to the main control module are incorporated into the structure in either a depth-first or breadth-first manner.

Bottom up Integration Bottom-up integration testing, as its name implies, begins construction and testing with atomic modules. Because components are integrated from the bottom up, processing required for components subordinate to a given level is always available and the need for stubs is eliminated.

3.3. System testing

System testing of software or hardware 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 System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that system elements have been properly integrated and perform allocated functions. Some of Different types of system testing are as follows:-

1. Recovery testing
2. Security testing
3. Graphical user interface testing
4. Compatibility testing

3.4. Acceptance testing

Acceptance testing (also known as user acceptance testing) is a type of testing carried out in order to verify if the product is developed as per the standards and specified criteria and meets all the requirements specified by customer. This type of

testing is generally carried out by a user/customer where the product is developed externally by another party. Acceptance testing falls under black box testing methodology where the user is not very much interested in internal working/coding of the system, but evaluates the overall functioning of the system and compares it with the requirements specified by them. User acceptance testing is considered to be one of the most important testing by user before the system is finally delivered or handed over to the end user. Acceptance testing is also known as validation testing, final testing, QA testing, factory acceptance testing and application testing etc. And in software engineering, acceptance testing may be carried out at two different levels; one at the system provider level and another at the end user level.

V. DISCUSSION

In this section difference between testing and debugging is shown. Software testing is a process that can be systematically planned and specified. Test case design can be conducted, a strategy can be defined, and results can be evaluated against prescribed expectations. Debugging occurs as a consequence of successful testing. That is, when a test case uncovers an error, debugging is the process that results in the removal of the error. The purpose of debugging is to locate and fix the offending code responsible for a symptom violating a known specification. Debugging typically happens during three activities in software development, and the level of granularity of the analysis required for locating the defect differs in these three. The first is during the coding process, when the programmer translates the design into an executable code. During this process the errors made by the programmer in writing the code can lead to defects that need to be quickly detected and fixed before the code goes to the next stages of development

VI. CONCLUSIONS

This paper on Software testing describes in detail about software testing, Software testing goals. Software testing is often less formal and rigorous than it should, and a main reason for that is because we have struggled to define best practices, methodologies, principles, standards for optimal software testing. To perform testing effectively and efficiently, everyone involved with testing should be familiar with basic software testing goals, limitations and concepts. Testing can show the presence of faults in a system; it cannot prove there are no remaining faults. System testing is the responsibility of a separate team. Integration testing is testing increments of the system; release testing involves testing a system to be released to a customer. Test

automation reduces testing costs by supporting the test process with a range of software tools.

REFERENCES

- [1] Sahil Batra and Dr. Rahul Rishi, "IMPROVING QUALITY USING TESTING STRATEGIES," *Journal of Global Research in Computer Science*, Volume 2, No.6, June 2011.
- [2] S.M.K Quadri and Sheikh Umar Farooq, "Software Testing-Goals, Principles and Limitations," *International Journal of Computer Applications*, Volume 6- No.9, September 2010.
- [3] Mohd. Ehmer Khan, "Different Forms of Software Testing Techniques for Finding Errors," *IJCSI International Journal of Computer Science Issues*, Vol. 7, Issue 3, No 1, May 2010.
- [4] Ajay Jangra, Gurbaj Singh, Jasbir Singh and Rajesh Verma, "EXPLORING TESTING STRATEGIES," *International Journal of Information Technology and Knowledge Management*, Volume 4, NO.1, January-June 2011.
- [5] Jovanovic and Irena, "Software Testing Methods and Techniques," *May 26, 2008*.
- [6] ger S. Pressman, "Software engineering: A practitioner's Approach," fifth edition, 2001.