



IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY



VOLUME : 1 ISSUE : 8 Print / Issue Publication Date: 09-Oct-2016



ISSN : 2455-2143



Indexed In



WWW.IJEAST.COM

editor@ijeast.com



SOFTWARE TESTING

Dimple Rajain
M.Tech Scholar, SKITM,
Bahadurgarh, India

Dr. V K Pandey
Asstt. Professor CSE, SKITM,
Bahadurgarh, India

Abstract - Software testing is a part of Software quality assurance process. Software quality assurance is concerned with software development process rather than just the documentations, coding and systems. Defects, error and bugs can creep in the software at any point of time, in any phase and in most unexpected way and have the capability to change the expected output of the software. It may be possible that a defect or bug may lead to complete failure or in giving output or unexpected output. So Software Testing becomes the essential phase for software development.

Keywords-testing, defects, failure, verification, validation, detecting error, black box testing, white box testing, gray box testing, principles of testing, sample life cycle of testing

I. INTRODUCTION

Software Testing is one of the most important phases of software development cycle. Software development cycle can be broadly divided into three categories:-

1. Development phase(Coding)
2. Testing phase
3. Maintenance

Software testing: Software testing is carried out to make certain that software is performing as expected to perform under all expected conditions and for expected inputs. Software testing is iterative process, when one bug is fixed, it can cause other bugs to disappear, or other or new bugs may appear. The idea of testing is to run the program with the intent to find possible errors.

Software testing has three objectives

1. verifying the software
2. finding errors(bugs)
3. validating the software

Difference between defects and failure

Defects/bugs means when mistake is done during building and coding of the software, failure means

when defects causes software to perform erratically. Defects in software may cause failure but not all defects cause failure. Failure may also occur due to failure in hardware, environment or error in execution of software.

Different phases in which mistake/error can be made by man

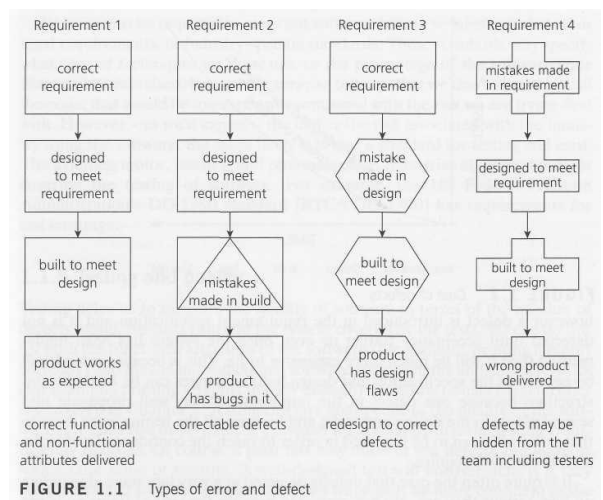


FIGURE 1.1 Types of error and defect

Source of image: Foundations-of-software-testing-certification

1. Verification: is the process to verify the system, i.e. whether system being built is right or wrong/ user requirements has been implemented in the product.

According to IEEE Standard Glossary of Software Engineering Terminology:

Verification is the process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

According to the ISO 9000 standard:

Verification is confirmed by examination and through provisions of objective



evidence that specified requirements have been fulfilled.

2. Detecting error: running the software or modules of software to find the bugs, errors, defects, that can cause software to run erratically. Main target of test team is greater test coverage with fewer tests and to find a defect at earliest possible location, the earlier a defect is found, the cheaper it is to fix it.

Static vs. dynamic testing

Static testing:-when the program/software to be tested is not executed, rather checked by inspection of code, or reviews or walkthroughs through code.

Static testing thus helps in doing verification.

Dynamic Testing: - When the program/software to be tested is executed with a given set of test cases to uncover bugs or errors. It can be started at any point of software development in order to test a particular section of code or function. Dynamic testing thus helps in validation.

Methods of detecting error

- Black Box testing- to test whether the software performs as expected.

The tester has no information about what is the code, how the code performs, which language is used for programming; he only knows the set of test cases and expected outputs for each test case.

The tester knows only what software is supposed to do, and have no knowledge about how it does.

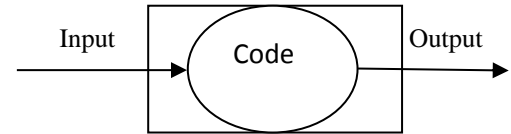


Black box testing

- White Box testing- test the logic/code of the software for possible error

The tester has full knowledge about the code, functions involved in the code and the language in which code is written, so he knows when a test case is run whether the code is performing as expected or not or in which part error is occurring.

The tester chooses inputs to exercise paths through the code and determine the appropriate outputs.



- Grey box testing – this testing method have some feature of both white box and black box testing method.

The tester have knowledge of internal data structures, code and programming language and what the code is supposed to do for the purpose of designing tests. The test case is executed at user level, i.e. concern is with right code are we getting right/expected output or not.

Grey box testing is useful in cases where the tests require modifying a back-end repository such as database, data warehouse or log file.

3. Validation: do deliverables satisfy the customer, is right software being built, is the software built and user specifications coherent to each other.

According to IEEE Standard Glossary of Software Engineering Terminology:

Validation is the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements.

According to the ISO 9000 standard:

Validation is confirmation by examination and through provisions of objective evidence that the requirements for a specific intended use or application have been fulfilled.

II. PRINCIPLES OF TESTING

There are seven principles of testing:-

1. Testing can never prove there are no defects: Testing is done to uncover the defects/bugs. Even after software is tested one cannot say that software is 100% error free i.e. testing can never be used to prove that there are no errors.
2. To check the software completely: One can never be sure to cover the whole and each aspect of the software, and also be sure have uncovered all the errors.



3. Testing time: Testing must be started as early as possible, earlier a defect is detected lesser will be the cost involved, and testing must be focused around objectives and requirement specifications.
4. A small number of modules contains most of the defects discovered during pre-release testing or shows the most operational failures.
5. The test cases should be reviewed regularly and new and different tests needs to be developed and executed on different parts of software or system to find more defects.
6. Modules/software performing different functions are to be tested differently, same test cases can't be used to test different functionality modules/Software.
7. If the system built is not of any use to customer then there is no use of finding and fixing the bugs.

III. A SAMPLE TEST CYCLE OF WATERFALL MODEL

- Requirement analysis- In this phase a tester is concerned with how efficiently are requirements of customer documented, so no loop holes are left on the ground of documentation of the requirements. During the design phase, testers determine which part of design are testable and on what parameters those tests work.
- Test Planning- As the name suggests this phase is concerned with planning of test, as many activities are carried out during tests, a plan is must, this may involve planning the strategy of plan, where to use what type of testing methodology, dividing the test team and assigning the areas on which they have to focus.
- Test development- Once planning has been done, now comes the most important thing of the testing, developing test procedures, test scenarios, test cases, test scripts, test datasets.
- Test Execution: After test cases and strategies have been developed these need to be executed on the software to find error/bug and then report any errors found to the development team to fix the bugs.

- Test Reporting- After testing a software, a testers make a report regarding the successfulness of the testing and whether the project is ready for release or not.
- Test result analysis/ Defect analysis - in this phase development team and client also have role to play, they collectively decide which defects need to be removed, fixed or deferred to be dealt later.
- Defect retesting- After a defect has been fixed by the development team the product needs to be tested again, to check whether the previous defect has been fixed or not, if any new defect has crept in while fixing a previous defect or triggered series of new defects if any new defect is found test execution is repeated again for that bug.
- Regression testing- a software is built of several modules, and from time to time new versions of software keep launching with additional functionalities, the testing of these added modules, functionalities is called regression testing and is done to ensure that the software product as a whole works correctly even after new updates.
- Test Closure- This is done after test has met the exit criteria specified in test planning, in this phase important learning from the tests, results, logs, documents related to project are saved and used for future reference.

IV. CONCLUSION

Since the software or project is built by humans, it's natural that mistakes can enter at any phase of software development therefore testing becomes important. The scope of testing is wide; the main aim of test team during testing is to cover all the program, with limited yet with range of test case inputs and to uncover maximum errors

Testing is not a single activity rather a process i.e. there is series of activities involved in the testing. Testing begins from the phase of requirement analysis of software development and continues till maintenance phase. There are different methods available for testing and testing must be done with utmost care and vigilance.



V. REFERENCES

1. Lessons learned in Software Testing, by C Kaner, J Bach and B. Pettichord
2. Testing Computer Software, by C. Kaner, J. Falk and H. Nguyen
3. Software Testing, by Ron Patton
4. Effective Software Testing by E. Dustin
5. IEEE Transactions on Software Engineering
6. [http:// www.engpaper.com](http://www.engpaper.com)
7. <http://www.cs.cmu.edu>
8. Foundations-of-software-testing certification

IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

ABOUT IJEAST

International Journal of Engineering Applied Science and Technology (IJEAST) is a peer-reviewed, open access journal that publishes high-quality research papers in the field of Engineering, Applied Science and Technology.

IJEAST aims to provide a platform for researchers, academicians, and professionals to share their innovative ideas, research findings, and practical experiences with the global scientific community.

FOCUS AREAS

- Engineering
- Applied Science
- Technology
- Innovation & Development
- Interdisciplinary Studies



PEER REVIEWED

All submissions are rigorously peer reviewed to ensure quality.



OPEN ACCESS

Free and unrestricted access to research for all.



GLOBAL REACH

Connecting researchers and professionals worldwide.



TIMELY PUBLICATION

We ensure a swift and efficient publication process.



For more information, visit our website

www.ijeast.com



INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

✉ editor@ijeast.com

🌐 www.ijeast.com

📍 India



2455-2143