

# TESTED SOFTWARE AND ITS RELIABILITY: AN EMERGING FIELD

<sup>1</sup>Varsha, <sup>2</sup>Rina Rani, <sup>3</sup>Shikha Tiwari

<sup>1</sup>Assistant Professor, <sup>2</sup>Assistant Professor, <sup>3</sup>Assistant Professor

<sup>1</sup>Computer Science Department

<sup>1</sup>Vaish Arya Kanya Mahavidyalaya, Bahadurgarh(Haryana), India.

**Abstract:** In paper, first we understand what is software engineering and why we can't use an un-engineered software. Software crisis, its causes, the present status and the possible solution to those crisis. Software engineering is one of the best option available to tackle software crisis as it integrates various activities (design, coding, testing and maintenance) and life cycle models to develop a bug free and reliable software. We also focus on how we get a bug free software using testing and then reliability is a need in software development life cycle.

**Keywords:** Software, testing, errors, SDLC, developer, tester.

**Software engineering:** Software engineering is a complete study of engineering to the design, development and maintenance of software. Software engineering was used to point out the issues of low-quality software projects. In a software, problems arise when it exceeds timelines, budgets, and reduced levels of quality. Software engineer ensures that the application is built consistently, correctly, on time and on budget and within requirements.

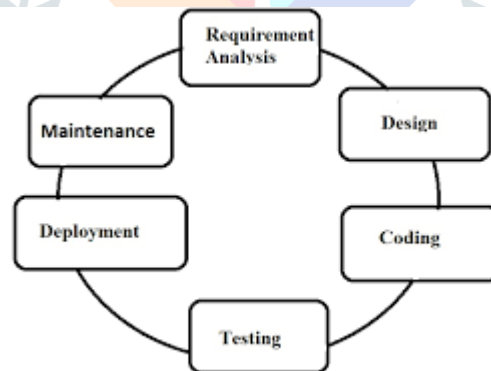


Fig 1: Software development life cycle

## 1. Software Crisis

There are lots of meaning of term software crisis. Let's see what are the software crisis in history and solution of those problems:

### 1.1 Inexperienced Developer:

Developers were not experienced that was a big issue in failure of software.

### 1.2 Failures of Big project:

Due to failure of big project huge amount of loss occurs like:

- 1.2.1** The North East blackout in 2003- has been major power system failures in the history of north which involves 100 power plants, 50 million customer faced problem, \$ 6million dollar financial loss.

- 1.2.2 Year 2000(Y2k)- refers to the obstacle in processing date after the Year 1999. In 1960-80 when shortened the four-digit date format like 1972 to a 2digit format like 72 because of that 2000 was shortened to 00. Million dollar were used up to handle this problem.
- 1.2.3 Arian- 5 Space Rocket: In 1996, developed at cost of \$7000 Million Dollars over a period of 10 years was destroyed within less than 1 minutes because of software bugs in rocket guidance system.

## 2. Software Testing

Software Testing performs a very important role in SDLC. Software testing is performed after the coding phase in SDLC.

**By whom Testing is done-** Those persons who are involved in software development they perform the testing. The various members of the professional team like: Software Tester, Project Manager, Software Developer and End Users performs the testing.

**When Testing should be stopped-** Testing the software is a continuous process. Developer can't claim that the software is 100% bug free, until testing is not performed.

### 2.1 SOFTWARE TESTING STRATEGIES:

There are various testing strategies that are being used for the purpose of testing:

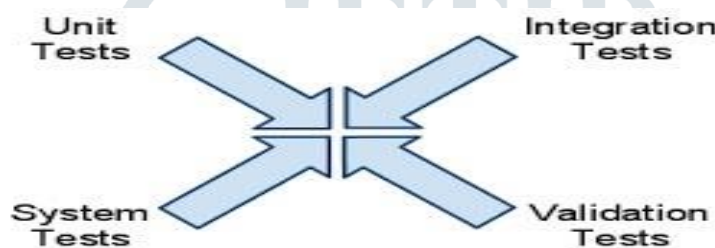


Fig 2: Software testing strategies

**2.1.1 Unit Testing-** Unit testing is performed at the lower level by the developers before it is moved to the team of testing to execute the test cases. If the output of any one of the module fails then the developers can't move this output as input for the next module.

**2.1.1.1 When the Unit testing is completed:** It is being completed before the **Integration Testing**.

**2.1.1.2 By whom Unit Testing is accomplished:** It is accomplished by software developer and their companion or very rarely by the independent Testers

**Popular Tools for Unit Testing are:** RSpec, Mocha, Tape and Jasmine, NUnit, JUnit.

**2.1.2 Integration Testing-** Integration Testing is performed immediately after the Unit Testing. In Integration Testing, all the modules are merged to determine whether they perform in a particular manner or not. If they perform in a particular manner then the testing is implemented on the modules. If any error is left in the Unit Testing then it can be again tested in integration testing so that no error is remained.

The basic intention of integration testing is to test how different parts of the system are grouped or work together. **For example**, a unit test for database access code would not be able to connect to a real database but the integration testing would.

**Testing is categorized into two parts:**

- (i) Top-Down Testing
- (ii) Bottom-Up Testing

**2.1.2.1 When the Integration Testing is completed:** It is being completed after **Unit Testing** and before **System Testing**.

**2.1.2.2 By whom Integration Testing is accomplished:** It is accomplished by either the independent developers or by the independent Tester.

**Popular mechanism for Integration Testing are:** TESSY, Steam, Mocha, Tape and Jasmine.

**2.1.3 System Testing-** This type of testing is organized to test the entire system. It is used to test the integrated component and the standard of quality.

**2.1.3.1 By whom System Testing is accomplished:** It is accomplished by the independent testers.

**For example:** When the pen is manufactured, the body, cap, ink cartridge are tested separately and then **unit testing** is performed. When more than two units are organized, they all are united and **integration testing** is performed. When the whole complete pen is concatenated then **system testing** is performed.

### 3. TESTING TECHNIQUES/ MECHANISM

There are various mechanism for testing the software:

1. Black Box Testing
2. White Box Testing

#### 3.1 BLACK BOX TESTING

This type of testing includes, the internal structure that are not accessible by the users. The software tester does not have any knowledge about the internal working of program. He just assumed the output value. These types of test can be functional or non-functional.

#### BLACK BOX TESTING APPROACH

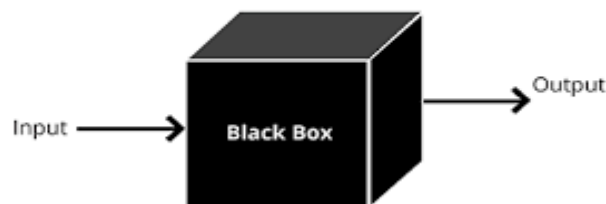


Fig. 3.1 Black box testing

Black Box Testing is named so because as we know that in the tester's eyes it is named black box but inner side no one sees. **Black Box Testing** is also known as **Functional testing, Specificational testing, Behavioral, Data Driven or Input-Output Driven.**

**For Example,** Without the recognition of the inner details of the website, we test the pages of web by the use of browser, authorize the input and then test and verify the outputs against the outcome that is expected.

**3.1.1 Equivalence Class Partitioning:** This type of technique divides the program input domain into the set of equivalence classes from where we can obtain the test cases.

**3.1.2 Boundary Value Analysis:** It is just opposite to partitioning the equivalence class because Boundary Value Analysis selecting the arbitrary input value to partition.

**For Example:** Programmer may inappropriately use  $\leq$  instead of  $<=$  for a function that calculates the square root of the integer value of the limit 0-7000.

**3.1.3 Cause Effect Graph:** This mechanism is used to identify the cases (Input conditions) and the effects (Output conditions).

**3.1.4 Comparison Testing:** Comparison testing is used for the fault tolerance in critical applications. In this a number of independent version of the software are developed for the similar specification.

### 3.2 WHITE BOX TESTING

In this type of testing, the internal structure/ details of the data item is known by or accessible to its user. Software tester is well aware about the internal working of program and test cases are made which are based on the code.

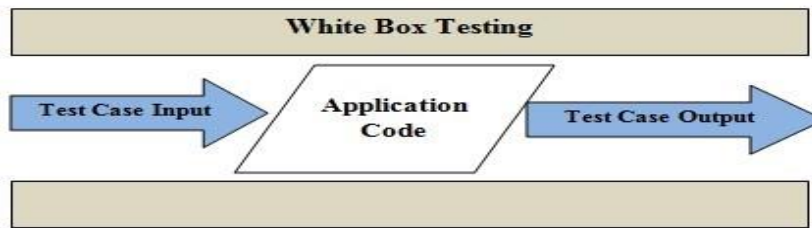


Fig. 3.2 White box testing

White Box Testing is named so because as we know that in the tester's eyes it is named white box and inner side everyone sees perfectly. **White Box Testing** is also known as **Glass Box, Structural; Clear Box, Open Box, Logic Driven, or Path Oriented.**

**Example:** Basically a tester and a developer identify the legal and illegal inputs and verify the output on the basis of expected outcome.

So, therefore we can say that white box testing is like the work of a mechanic who only needs to know why the car is not working correctly.

### 4. Software Reliability

One of the major reason of Product Unreliability is the presence of system's error. Software does not get mature but due to the presence of errors or faulty design software unreliability is occur. Reliability is considered the most essential characteristics that inherent from the concept "software quality". Software reliabilities depend upon the how well the software functions fulfill the requirements of the customer. The life cycle of software includes many test items such as documents, manuals, reports, plans, code configuration data and test data which help in evaluating its reliability. "Fig 4" shows the relationship between Failure rate and Time as shown below:

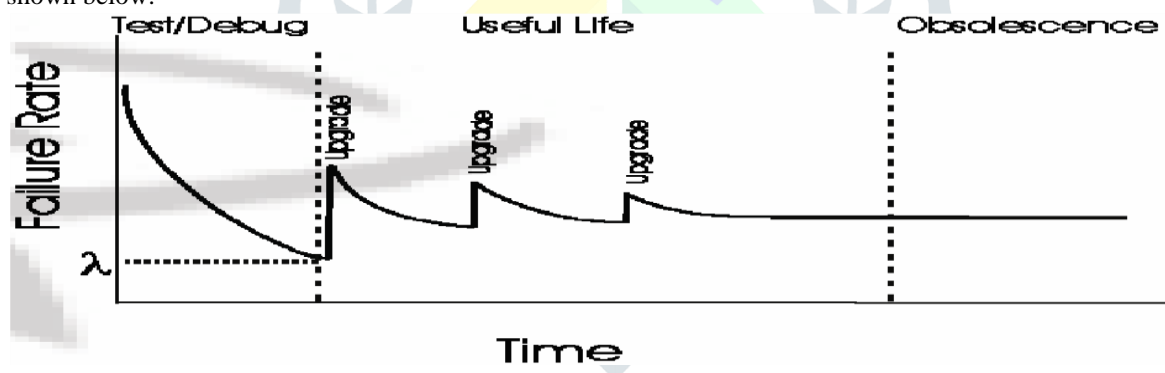


Fig 4: Time Vs Failure Rate Graph

### Characteristics of Software Reliability

#### 4.1 Failure occurs primarily due to design faults:

For detecting the error, Design is modified for repairs to make it powerful against conditions

#### 4.2 There is no wear-out phenomenon:

- 1) Software bugs occur without any warning.
- 2) While doing reforms, "Old" code can result in more number of failure rate because of errors.
- 3) External environment conditions generally not affect the reliability of the software.
- 4) Internal environment related conditions, such as inappropriate clock speeds or insufficient memory affect software reliability.

### 4.3 Reliability is not time dependent

- 1) Failure happens due to the error prone execution
- 2) The growth of the reliability is observed as errors are detected and corrected.

## 5. Software Reliability Activities

The software reliability process includes software development, operations, and maintenance. A software reliability process includes faults, defects, corrections, errors, updating, and expenses on the resource, such as manpower effort. Some of the Reliability activities are as follows:

**5.1 Construction:** Generation of new documentation and code artifacts

**5.2 Combination:** It forces on reusability of old documents and code components with the new one.

**5.3 Correction:** Analyzing and removing document and code related defects by analyzing the test items.

**5.4 Preparation:** Generating of different test items.

**5.5 Testing:** Test cases are executed, to know the trigger points where failure occurs frequently.

**5.6 Identification:** Categorized each error or bug whether new or previously

**5.7 Repair:** Faults are removed which possibly introduces new faults for which regression testing is done.

**5.8 Validation:** Perform checks to make sure that repairs are effective and have not affected other parts of the software.

**5.9 Retest:** implementation of the cases to check for specified repair's completion. If it is incomplete, new test cases may be needed to repair them further.

## Conclusion

Software engineering is first need in developing software in today's life. It may cause loss of lot of money if we use product that is not engineered. Software testing is done with the intention of finding errors. A lot of money spent in testing phase of SDLC. Software testing must be done properly in order to avoid destroy of whole project. Hardware aged and rust with time but software doesn't. Software reliability is very necessary in order to use software a long period of time.

## References

- [1] <https://economictimes.indiatimes.com/definition/software-engineering>
- [2] <https://www.castsoftware.com/glossary/importance-of-software-engineering-code-of-ethics-future>
- [3] <http://technav.ieee.org/tag/1579/software-testing>
- [4] Chauhan, R; Singh, I; (2014) "Latest Research and Development on Software Testing Techniques and Tools", <http://inpressco.com/wp-content/uploads/2014/07/Paper122368-2372.pdf>.
- [5] Yadav, P; Kumari, P; (2015) "REVIEW PAPER ON SOFTWARE TESTING", [http://www.ijirt.org/vol1/paaperpublished/IJIRT102003\\_PAPER.pdf](http://www.ijirt.org/vol1/paaperpublished/IJIRT102003_PAPER.pdf).
- [6] Aasia Quyoum, Mehraj-Ud - Din Dar, S. M. K. Quadri, "Improving Software Reliability using Software Engineering Approach" Volume-5 Number-3
- [7] Ellen Walker, "Applying Software Reliability"
- [8] [https://www.google.com/search?rlz=1C1AWFC\\_enIN741IN741&tbm=isch&sa=1&ei=1iCLXNfxNsOBvgTrtrqICQ&q=black+box+testing+with+black+and+white+images&oq=black+box+testing+with+black+and+white+images&gs\\_l=img.3...12462.18591..18798...0.0..2.778.6863.0j15j5j2j1j1j2.....1...1..gws-wiz-img.....0i8i30j0i24.AzavahqbLaQ](https://www.google.com/search?rlz=1C1AWFC_enIN741IN741&tbm=isch&sa=1&ei=1iCLXNfxNsOBvgTrtrqICQ&q=black+box+testing+with+black+and+white+images&oq=black+box+testing+with+black+and+white+images&gs_l=img.3...12462.18591..18798...0.0..2.778.6863.0j15j5j2j1j1j2.....1...1..gws-wiz-img.....0i8i30j0i24.AzavahqbLaQ)

[9] [https://www.google.com/search?rlz=1C1AWFC\\_enIN741IN741&biw=1366&bih=657&tbm=isch&sa=1&ei=WjCLXNSDH8ew9QPj24LYDg&q=sdlc+black+and+white+images&oq=sdlc+black+and+white+images&gs\\_l=img.3...2979.2979..3470...0.0..0.120.120.0j1.....1....1..gws-wiz-img.cQf67X8MHVI](https://www.google.com/search?rlz=1C1AWFC_enIN741IN741&biw=1366&bih=657&tbm=isch&sa=1&ei=WjCLXNSDH8ew9QPj24LYDg&q=sdlc+black+and+white+images&oq=sdlc+black+and+white+images&gs_l=img.3...2979.2979..3470...0.0..0.120.120.0j1.....1....1..gws-wiz-img.cQf67X8MHVI)

### Biography

Hi! My name is Varsha Assistant Professor(NET qualified) in Vaish Arya Kanya Mahavidyalaya, Bahadurgarh, Haryana, India. I had done MCA (Masters in Computer Application). My area of research is Software Testing and Reliability. I have completed my Master of Computer Application's degree in 2013 from Maharishi Dayanand University with 86% marks (5<sup>th</sup> rank in M.D.U). I have attended an International Conference and presented a paper on "Globalization in E-Commerce".

