

# Qualitative Analysis of Various Levels of Software Testing

Amit Kumar

Assistant Professor, Department of Computer Science & Engineering, Maulana Azad College of Engineering & Technology, Patna, Aryabhata Knowledge University (AKU), India

**Abstract:** Software testing is very important activity in Software Development Life Cycle (SDLC). It is one of the important activity which needs lots of time and effort. Various testing techniques are used to find errors in the software. Testing is involved at various stages of software development like unit testing, integration testing, system testing, acceptance testing etc. Different testing techniques namely Black Box Testing, White Box Testing used to test software.

## 1. Introduction

Software testing is used to find the errors generated within a software during its development. A well tested Software system is necessary for the quality of a Software. Conducted to ascertain the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Testing techniques include the process of executing a program or application with the intent of finding software errors. The testing of software is an important means of assessing the software to determine its quality. Since testing typically consumes 40 to 50% of development efforts, and consumes more effort for systems that require higher levels of reliability, it is a significant part of the software engineering. Testing is very important area of computer science. There are generally four recognized levels of testing: unit/component testing, integration testing, system testing, and acceptance testing. Tests are frequently grouped by where they are added in the software development process, or by the level of specificity of the test. Below you can see 4 different levels of testing:

- 1) Unit Testing
- 2) Integration Testing
- 3) System Testing
- 4) Acceptance Testing

### Unit Testing

UNIT is individual Testing defined as a type of software testing where units/ components of software are tested. Unit Testing of software applications is done during the development (coding) of an application. The objective of Unit Testing is to isolate a section of code and verify its correctness. In procedural programming, a unit may be an individual function or procedure. Unit Testing is usually performed by the developer. In SDLC, STLC, V Model, Unit testing is first level of testing done before integration testing. Unit testing is a White Box testing technique that is usually performed by the developer. Though, in a practical world due to time crunch or reluctance of developers to tests, QA engineers also do unit testing. Sometimes software developers attempt to save time by doing minimal unit testing. This is a myth because skipping on unit testing leads to higher Defect fixing costs during System Testing, Integration Testing and even Beta Testing after the application is completed. Proper unit testing done during the

development stage saves both time and money in the end. Here, are key reasons to perform unit testing.

- Unit Tests fix bug early in development cycle and save costs.
- It helps understand the developers the code base and enable them to make changes quickly
- Good unit tests serve as project documentation.
- Unit tests help with code re-use. Migrate both your code and your tests to your new project.

### Advantages of Unit Testing

- Developers looking to learn what functionality is provided by a unit and how to use it can look at the unit tests to gain a basic understanding of the unit API.
- Unit testing allows the programmer to refactor code at a later date, and make sure the module still works correctly (i.e. Regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified and fixed.
- Due to the modular nature of the unit testing, we can test parts of the project without waiting for others to be completed.

### Disadvantages of Unit Testing

- Unit testing can't be expected to catch every error in a program. It is not possible to evaluate all execution paths even in the most trivial programs
- Unit testing by its very nature focuses on a unit of code. Hence it can't catch integration errors or broad system level errors.

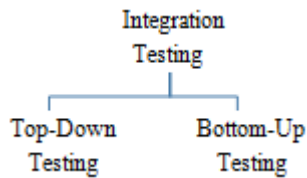
### Integration Testing

Integration Testing is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. Integration Testing focuses on checking data communication amongst these modules. After integrating two different components together we do the integration testing. As displayed in the image below when two different modules 'Module A' and 'Module B' are integrated then the integration testing is done. Integration testing is done by a specific integration tester or test team. Integration testing follows two approach known as 'Top Down' approach and 'Bottom Up' approach as shown in the image below:

Volume 8 Issue 7, July 2019

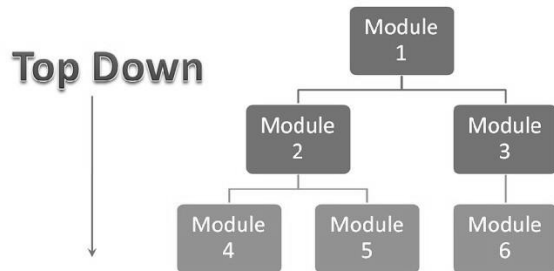
[www.ijsr.net](http://www.ijsr.net)

Licensed Under Creative Commons Attribution CC BY



### Top-down Integration Testing

Testing takes place from top to bottom, following the control flow or architectural structure (e.g. starting from the GUI or main menu). Components or systems are substituted by stubs. Below is the diagram of 'Top down Approach':

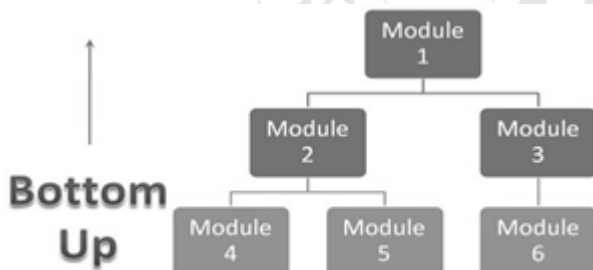


### Advantages

- The tested product is very consistent because the integration testing is basically performed in an environment that almost similar to that of reality.
- Stubs can be written with lesser time because when compared to the drivers then Stubs are simpler to author.

### Bottom up Integration Testing

Testing takes place from the bottom of the control flow upwards. Components or systems are substituted by drivers. Below is the image of 'Bottom up approach':



### Advantage of Bottom-Up approach:

- In this approach development and testing can be done together so that the product or application will be efficient and as per the customer specifications.

### Disadvantages of Bottom-Up approach:

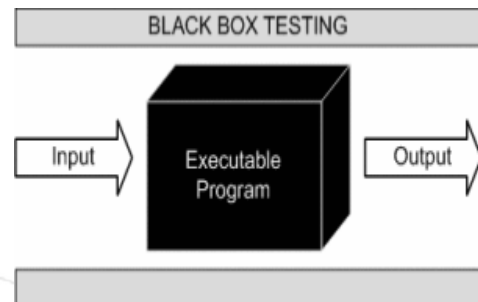
- We can catch the Key interface defects at the end of cycle
- It is required to create the test drivers for modules at all levels except the top control

### System Testing

System Testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. System Testing (ST) is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested

from an end-to-end perspective. System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing.

Black Box Testing, also known as Behavioral Testing, is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.



This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Behavior or performance errors
- Initialization and termination errors

Testing involves System testing the software code for following

- Testing the fully integrated applications including external peripherals in order to check how components interact with one another and with the system as a whole. This is also called End to End testing scenario.
- Verify thorough testing of every input in the application to check for desired outputs.
- Testing of the user's experience with the application.

### Acceptance Testing

Acceptance Testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery. User acceptance testing (UAT) is the last phase of the software testing process. During UAT, actual software users test the software to make sure it can handle required tasks in real-world scenarios, according to specifications. ... UAT is also known as beta testing, application testing or end user testing.

### Prerequisites of User Acceptance Testing:

Following are the entry criteria for User Acceptance Testing:

- Business Requirements must be available.
- Application Code should be fully developed
- Unit Testing, Integration Testing & System Testing should be completed
- No Showstoppers, High, Medium defects in System Integration Test Phase -

- Only Cosmetic error is acceptable before UAT
- Regression Testing should be completed with no major defects
- All the reported defects should be fixed and tested before UAT
- Traceability matrix for all testing should be completed
- UAT Environment must be ready
- Sign off mail or communication from System Testing Team that the system is ready for UAT execution

#### Types of User Acceptance Testing

- Alpha & Beta Testing
- Contract Acceptance Testing
- Regulation Acceptance Testing
- Operational Acceptance Testing

#### Alpha & Beta Testing

Alpha Testing normally takes place in the development environment and is usually done by internal staff. Long before the product is even released to external testers or customers. Also potential user groups might conduct Alpha Tests, but the important thing here is that it takes place in the development environment.

Based on the feedback – collected from the alpha testers – development teams then fix certain issues and improve the usability of the product.

Beta Testing, also known as “field testing”, takes place in the customer’s environment and involves some extensive testing by a group of customers who use the system in their environment. These beta testers then provide feedback, which in turn leads to improvements of the product.

Alpha and Beta Testing are done before the software is released to all customer.

#### Contract Acceptance Testing

Contract Acceptance Testing means that a developed software is tested against certain criteria and specifications which are predefined and agreed upon in a contract. The project team defines the relevant criteria and specifications for acceptance at the same time when the team agrees on the contract itself.

#### Regulation Acceptance Testing

Regulation Acceptance Testing, also known as Compliance Acceptance Testing, examines whether the software complies with the regulations. This includes governmental and legal regulations.

#### Operational acceptance testing

Also known as Operational Readiness Testing or Production Acceptance Testing, these test cases ensure there are workflows in place to allow the software or system to be used. This should include workflows for backup plans, user training, and various maintenance processes and security checks.

## 2. Conclusion

Testing is a process to evaluate the quality of software. It is an effort intensive activity. Various types of testing are used to test software. There is scope for automation in the activities of testing but testers experience is very much important for successful testing. Software testing is component of software quality control. Various Levels of testing are used to detect errors in the software like unit testing, integration testing, acceptance testing, system testing are used to test a system. percentage of budget should be invested on Software Testing because a well tested software is guarantee of a good quality. This is the reason that Software Testing is the most crucial phase of Software Development Life Cycle (SDLC).

## References

- [1] [https://en.wikipedia.org/wiki/Software\\_testing](https://en.wikipedia.org/wiki/Software_testing)
- [2] Lu Luo, “Software Testing Techniques Technology Maturation and Research Strategy”, Class Report for 17-939A.
- [3] Abhijit A. Sawant, Pranit H. Bari, P. M. Chawan, “Software Testing Techniques and Strategies”, International Journal of Engineering Research and Applications, Vol. 2 Issue 3, May-Jun 2012, pp.980-986
- [4] SMK Qadri et. al, “Software Testing – Goals, Principles, and Limitations”, International Journal of Computer Applications, Volume 6– No.9, September 2010
- [5] <http://www.cs.umd.edu/~mvz/cmsc435-s09/pdf/slides10.pdf> [6] Jovanović Irena, “Software Testing Methods and Techniques