

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Paper / Case Study

Available online at: www.ijarcsms.com

Comparative Study on Integration Testing and System Testing

Shikha Verma

Delhi – India

Abstract: In the context of software engineering, system testing and integration testing exercises a software system's coexistence with others. This research paper is based on the comparative study on Integration and System Testing. The purpose, approaches, aims, expectations and types of these testing is discussed in this paper

Keywords: Testing, Testability, Aspects of testing, System testing, Integration testing, alpha testing, incremental approach, Acceptance testing, Beta testing.

I. INTRODUCTION

Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Testing a product is relatively independent of the development method used to develop it. The purpose of testing is to find faults, being a destructive process it is crucial to be performed. It enhances the integrity of the system by detecting deviations in design and error in the systems.

II. TESTING

Definition: - Testing is the process of examining something with the intention of finding faults it may reveal symptom of error but it may not uncover the precise cause of that error. The idea of testability is the single crucial factor behind software testing.

ASPECTS OF TESTING- Testing is very difficult, complex and challenging activity. That is why programmer should always think about testability of a program right from the beginning.

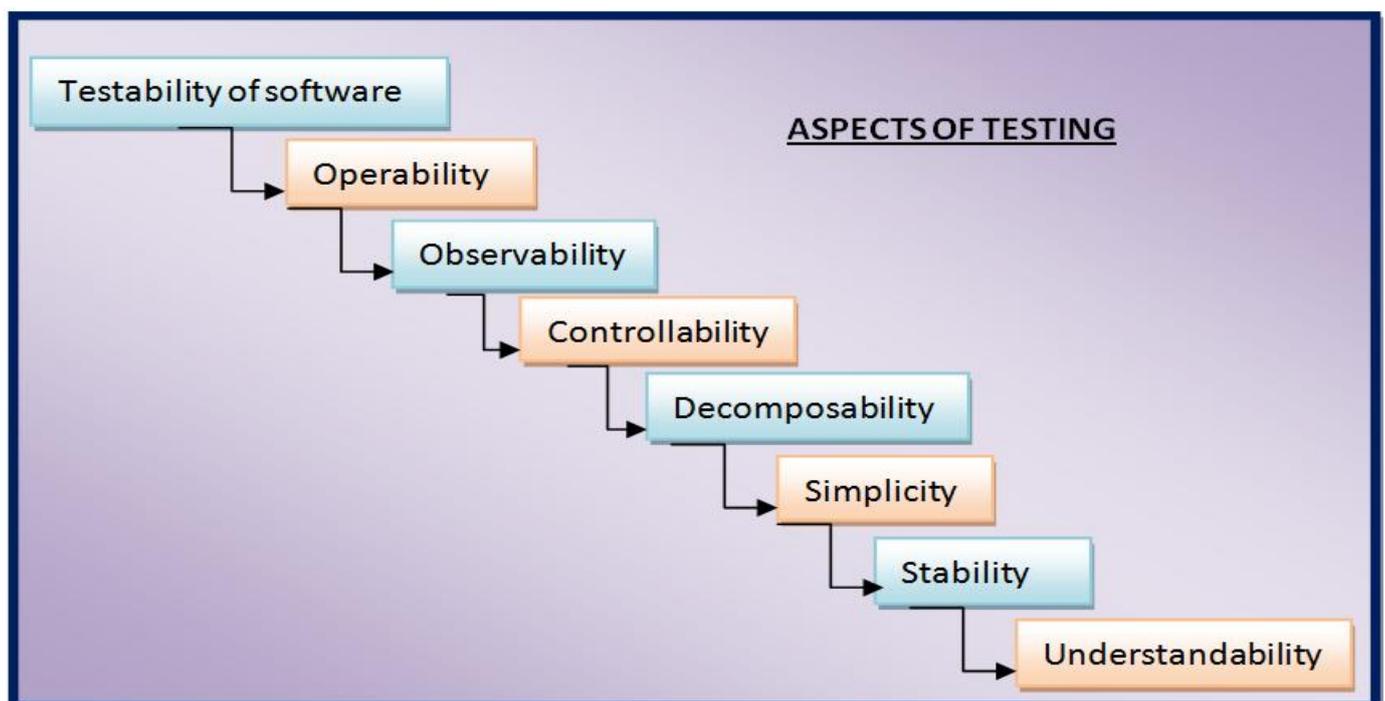


Fig 1. Various aspects of software testing

Aspects of Testing

Operability –	The more correct the program, the simpler it is to test.
Observability-	It is easy to verify the output of testing.
Controllability-	More control in the software development process means more optimized and automated testing.
Decomposability-	It should be possible to control the scope of testing to identify and remove problem quickly.
Simplicity-	The lesser things we need to test for is the better.
Stability-	The changes to software should be minimum, coordinated and controlled.
Understandability –	More precise the documentation, easier it is to test.

Overview of Integration Testing

Fig 2. Integration testing

- The integration testing focuses on finding defects which mainly arise because of combining various components for testing
- Main objective is to take unit tested components and build a program structure.
- Integration testing is done by developers / Quality assurance teams
- These members tests both normal processing and exceptions
- Errors can be:-
 - ✓ Interface incompatibility
 - ✓ Incorrect parameter values
 - ✓ Resources problems
 - ✓ Run time exceptions

Integration Testing Strategy

- The entire system is viewed as a collection of subsystems (sets of classes) determined during the system and object design.
- Assumption: System Decomposition is hierarchical

- The order in which the subsystems are selected for testing and integration determines the testing strategy
 - Big bang integration (No incremental)
 - Bottom up integration
 - Top down integration
 - Sandwich testing
 - Variations of the above
- For the selection use the system decomposition from the System Design

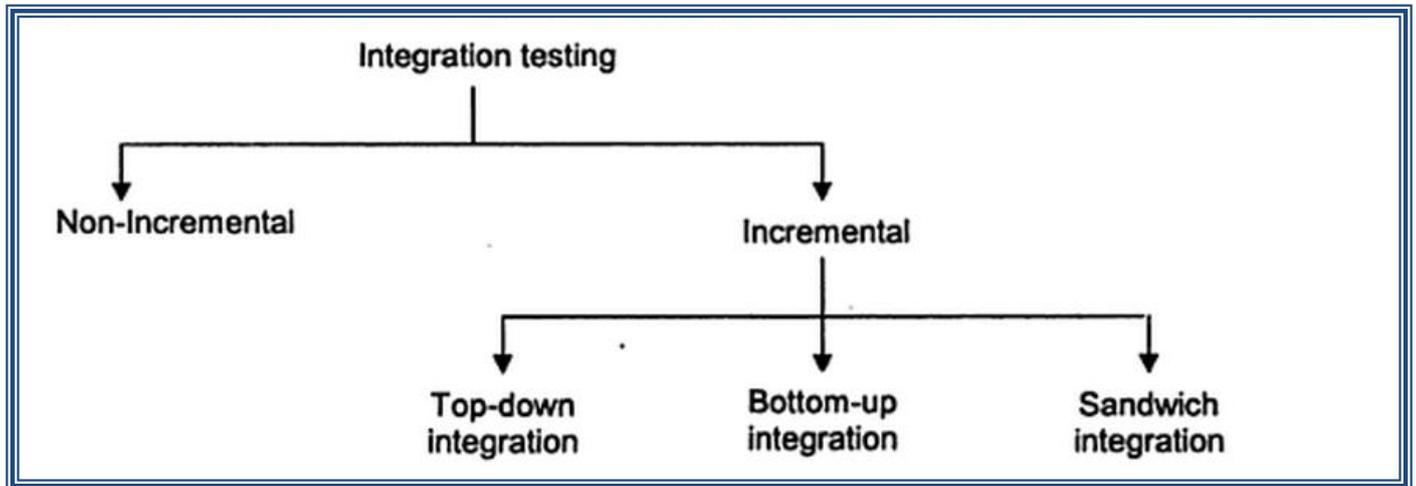


Fig 3. Types of integration testing

Overview of System Testing

- System testing is one of most important phase in testing process.
- This phase starts after the completion of phases like Unit Testing and Integration testing.
- System testing is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.
- System testing is the performed on completely developed and integrated system.
- Once complete system has been built then it has to be tested against system specifications to verify whether it delivers the required features and quality standards.
- System testing falls within scope of Black box testing and as such, should require no knowledge of the inner design of the code or logic.



Fig 4. System Testing

- System testing verifies the entire product, after integrating all software and hardware components and validating/testing it according to project requirement specifications.
- As a rule, System testing takes, as its input, all of the "integrated" software components that have successfully passed Integration testing and also the software system itself integrated with any applicable hardware system.

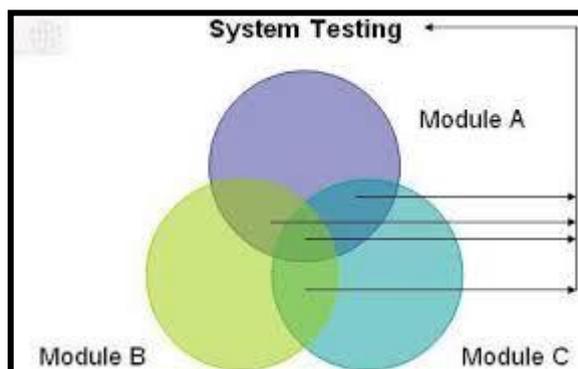


Fig 5. System testing performed on module A, B, C

System testing comprises of two types of testing: Functional testing and Non-Functional testing.

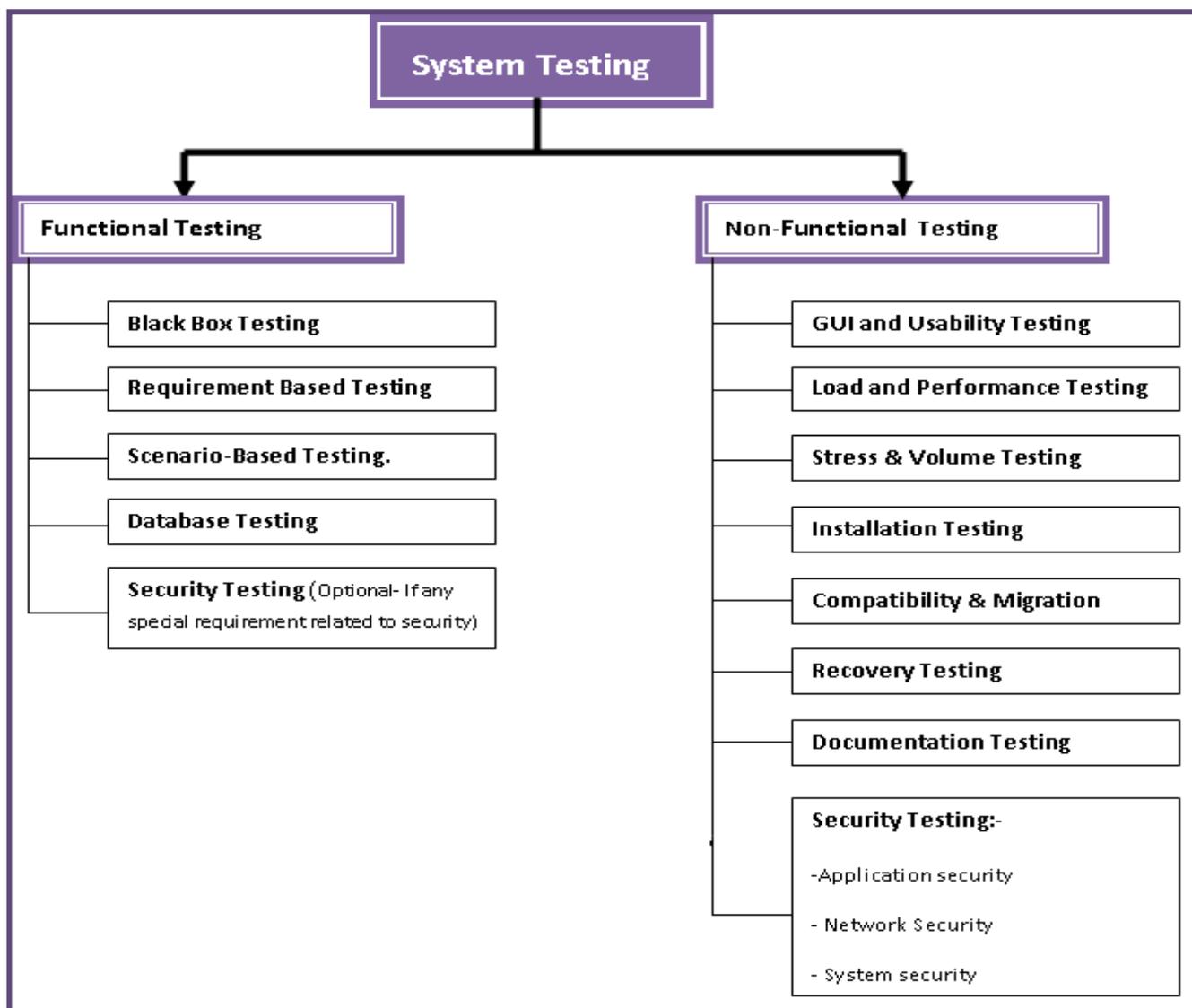


Fig 6. Types of System Testing

TABLE I
COMPARATIVE ANALYSIS OF INTEGRATION AND SYSTEM TESTING

	<u>Integration Testing</u>	<u>System Testing</u>
Technique	It is a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing.	The system testing process is concerned with finding errors that result from unanticipated interaction between subsystem and system component.
Aim	It involves integrating independent software unites or components to form a sizeable build and then testing the assembly.	It involves integration the subsystem to make up the entire system.
Purpose	To prove that all areas of software units or components interface with each other and also to verify the functionality that there are no gaps in the dataflow.	Verifying end-to-end work flows and scenarios. All the software components, all the hardware components, all internal interfaces, all the external interfaces should be tested.
Environment	Integration testing takes place either in the development environment or in test environment using real data, if possible else simulated data need to be created to model real data.	System testing requires system test environment that comprise of deployment like environment from hardware and software requirements.
Expectations	The primary emphasis is verification of each component and inter-modular interfaces.	The primary emphasis is verification of the system as a whole.
	It tries to test all testable requirements at least once by the end of testing	This serves as a final verification of requirements and design.
	Hardware specification should be verified for correctness and compliance with specification.	Correct operation of external interface must be verified and some performance test may be conducted and used to model or extrapolate behavior.
Considerations	Integrating independent software units or components to form a sizeable build and then testing the assembly	Perform functional test, Regression tests, Performance tests, Load test.
	To find any issues in interface among units or components	Perform interface validation tests
	To find any gaps in the data flow	Perform security test
Tasks	Integrate software units or components	Arrive at detailed system test plan
	Prepare integration test report	Perform system testing report
Approaches	Incremental Approach	Alpha testing
	Top-down approach	Beta testing

	Bottom up approach	Acceptance testing
	Regression testing	
	Smoke testing	
	Sandwich integration	
Diagrammatical Representation		

III. CONCLUSION

This paper studies about the SOFTWARE TESTING, their ASPECTS, INTEGRATION AND SYSTEM TESTING .The purpose of integration testing is to prove that all areas of software units or components interface with each other and also to verify the functionality that there are no gaps in the dataflow while in System testing main aim is concerned with finding errors that result from unanticipated interaction between subsystem and system component. Both these methods of testing have their own role and importance in the lifecycle of software and their testing.

References

1. BRITISH COLUMBIA – System Testing Process
2. Book -Software Testing and Quality Assurance
3. http://www.kabinfol.net/kabinfol_SystemTesting.asp
4. <http://ix.cs.uoregon.edu/~michal/Classes/W98/LecNotes/10-Testing-system.pdf>
5. <http://blog.teamgrowth.net/index.php/quality-assurance-testing/introduction-to-system-testing>
6. <http://www.cs.bham.ac.uk/~exc/Teaching/STesting/Lectures/14.%20System%20testing.pdf>

AUTHOR(S) PROFILE



Shikha Verma Completed the MCA degree in Computer Applications from Guru Gobind Sing Indraprastha University and received B.A degree in Computer Applications from Delhi University in 2013 and 2010, respectively.