

# SE3-Master Report

## **A Comparison and Contrast between Unit Testing and Integration Testing**

Student Name : Abdulaziz Almaghrabi

Student ID number : 2101607

Fan number : alma0107

Date: 30/6/2014.

## Table of Content

|   |          |
|---|----------|
| <b>Introduction .....</b>   | <b>3</b> |
| <b><i>A Comparison and Contrast between Unit Testing and Integration Testing.....</i></b> | <b>3</b> |
| <b><i>Conclusion.....</i></b>   | <b>6</b> |
| <b>List of References.....</b>  | <b>7</b> |

## **Introduction**

Software development is accomplished through different modules which perform different functions such as data input, initializing variables as well as performing calculations and generating reports. Testing each module is an essential function software engineering, and thus should be performed throughout the development process (James 2008). Although testing is generally performed when software development process is nearly at its end, this is considered a wrong practice as it is perceived that performing testing in later stages results in consuming more time and money to correct errors. Therefore, different levels of testing such as unit testing, system testing, integration testing and acceptance testing are crucial in software development (Khurana 2009).

Research shows that Information Technology managers and professionals may have different opinions regarding software developments principles. However, most of them support the principle that the software they deliver need to be accurate and reliable. Recent surveys about software development managers, reveals that software testing and quality assurance has become the top most issue as testing is commonly confused to mean quality assurance yet the two aspects have different functionality roles. It cannot be denied that a well-tested product that was wrongly conceived and incompetently designed and programmed will finally become a well-tested but bad product. On the other hand, software testing is highly considered as one of the core technical activities undertaken to enhance the quality of software. The main objective of quality assurance in software development is to ensure that the software delivered minimize defects as well as meets the users' requirements or specified levels of functionality, reliability and performance (Iacob & Constantinescu 2008, p.241-49).

### **A Comparison and Contrast between Unit Testing and Integration Testing**

Unit testing is important in checking independent units or simply the modular design of a software being developed. Successful unit testing is followed with integration testing to check the integration condition of various units or the physical design of the software. Once integration testing is complete, system testing is performed purposely to verify the logical design of the

software prior to performing the acceptance testing to check if the software meets the user requirements (Khurana 2009, p.224).

Basically, unit testing is an easy process and it involves testing each module of the software which saves effort, time and money spent in integration testing or compared to other testing done on the finished product. Research shows that performing unit tests enhances the performance of the software or system developed. Once the different units of software are found to be functioning properly, integration testing is performed to integrate the components to ensure that proper integration the final product is made. In particular, the integration testing is accomplished to detect design errors and to test the interfaces between the various modules involved in the development of the software. In software development projects where object-oriented model is applied, integration testing is performed to test the different classes to determine their proper integration. However, in component object model, this type test is done on different components to determine their ability for integration. In the case of Web Services, integration testing means testing the effectiveness and proper communication links between the service and web page (James 2008, p.234).

Unit testing is different from integration testing because it is a function performed to test the individual units of software being developed. The fact that software comprises of various units or modules, it means that detecting errors within the units becomes a simple task if unit testing performed. In addition, less time is consumed because units checked are small in size. However, it is important to note that incorrect output generated by a single unit if used as input to the next unit can lead to wrong output. This implies that ignoring to correct such a process will cause the entire software to produce the unexpected user requirements. Since many of the Off-the-shelf components are also used in software development but cannot be unit tested, integration testing plays an integral role in such circumstances where unit testing is not applicable (Khurana 2009, p.225).

Integration tests make sense in most systems in two main configurations; the first one is when it is written as extra layer of regression that is used test beyond unit tests. Secondly, if applied in legacy code situations but written before the unit tests. Being used as additional regression test,

integration tests can be performed particularly when the entire system has to be tested to ensure that deployment of each system component works properly. Therefore, integration test is totally different from a unit test but more related to performance, system and acceptance tests. Overall, through integration testing, it becomes easy to measure the quality of software based on the identified defects particularly for functional and non-functional software requirements as well as the associated characteristics such as efficiency, maintainability, reliability and usability (Gilberfeld 2011).

It is important to note that unit testing is sometimes referred to as Component testing performed to ensure that reliable unit is compatible or meet the expected requirements. Unlike integration test which involves performing tests on an integrated system, Unit testing is done on each unit in isolation to identify the various inputs to a unit and the expected outputs for the same process. Specifically, Component testing or unit test is performed by the programmer and not by testers because it involves applying detailed knowledge of how the internal program design and code functions. However, integration test involves as well other experts or testers who have undertaken a role in the development of software. In general, integration testing is done on integrated modules with the main objective of verifying the combined functionality once the system has been integrated. Thus, integrated testing is accomplished on software modules such as code modules, individual, client and server applications on a given network platform (Dasari 2010).

With integration testing, the entire system is regarded as a collection of subsystems or set of classes that are specified during the system and object design. Like the unit testing which tests individual units, integration testing also verifies the interfaces between subsystems and their interactions. However, the strategy used determines the order through which such subsystems are selected conduct testing and integration of the software or system. It is notable that integration testing is done because unit tests is limited to testing the unit in isolation and a number of failures occur as a result of the interaction of subsystems. Therefore, without conducting integration testing it means that the system test will be a more time consuming task because the failures that could be discovered during the integration testing will surface after deploying the system and can lead to huge costs (Bruegge & Dutolt 1999).

## **Conclusion**

Software testing is a core technical activity undertaken to enhance the quality of software. Unit testing is crucial in checking independent units or the modular design of a software being developed because performing unit tests improves the performance of the software or system developed. Integration testing is performed only if the different units of software are found to be functioning properly purposely to integrate the components to ensure the proper integration of the final product. The main difference between unit testing and integration testing is that integration test involves performing tests on an integrated system, while Unit testing is done on each unit in isolation to identify the various inputs to a unit and the expected outputs for the same process. In general, quality assurance in software development is crucial in ensuring that the software delivered minimizes defects and meets users' requirements or the specified levels of functionality, reliability and performance. Therefore, both unit testing and integration testing are important in delivering accurate and reliable software.

## List of References

Bruegge, B & Dutoit, H.A, 1999, Chapter 11, *Testing, and Part 2: Integration and System Testing: Object-Oriented Software Engineering using UML, Patterns and Java*, University of Georgia.

Dasari, S., 2010, '*The Difference between Unit Testing, Assembly Testing, Integration Testing and Regression testing*', Retrieved June 14, 2014 from, <<http://www.aspdotnet-suresh.com/2010/09/whats-difference-between-unit.html>>

Iancob, I.M & Constantinescu, R., 2008, 'Testing: First Step towards Software Quality, *Journal of Applied Quantitative Methods*', Vol.3, No.3, pp.241-253.

James, K.L, 2008, *Software Engineering*, PHI Learning Pvt. Ltd.

Khurana, R., 2009, *Software Engineering, 1E*, Vikas Publishing House Pvt Ltd.

Zilberfeld, G, 2011, "*The Difference between Unit Testing and Integration Testing*", SYS-CON Media, Inc, Retrieved June 14, 2014 from, <<http://www.typemock.com/unit-tests-integration-tests>>