**Software Testing**

**Thurman Peele**

**CST 499 Capstone for Computer Software Technology**

**Amjad Alkilani**

**June 21, 2020**

**Software Testing**

Software testing is seen as the process to look at the functionality of a software application. The reason to look at the functionality of a software application is to find whether the developed software met the specified requirements or not and to identify any defects to ensure that the product is defect free so that a quality product is produced. Idrus states “Software testing is a knowledge intensive phase within software development and software engineering fields as vast amount of knowledge is captured, used, shared, stored and reproduced throughout the testing activities. It involves the process of identifying software errors, bugs or failures that need to be fixed before installation in the live environment and heavily relies on business process and system knowledge” (Idrus & Ali, 2019). Software inspections will help greatly because it’ll reduce the downtime if it was to go down. It’ll help prevent any outages. You will be able to discover the program defects. By testing the software, you make sure the software meets the requirements for the customer. Also, during testing errors can mask other errors. The inspection is a static process, so you don’t have to be concern with interactions between errors. Incomplete versions of a system can be inspected without additional cost. If a program is incomplete, then you need to develop specialized test harnesses to test the parts that are available.

There are different levels to software testing, and they are unit testing, integration testing, system testing, and acceptance testing. Each of the below four testing levels verifies the software functionality for correctness, quality, and performance.

Unit testing is used to validate that each unit of the software performs as designed. An example of unit testing can be testing a function whether a statement or loop in a program is functioning properly or not then this is entitled as unit testing. Tsui states “Unit testing is a process followed by programmer to test each piece or unit of software” (Tsui, 2018). Each unit is tested separately and the main goals of unit testing is to isolate each and every section of code, save of testing cost, make sure individual parts are correct, find bugs in the early stages of the development cycle, and to allow developers to refactor of upgrade code at a later time. One advantage of unit testing is that it simplifies the debugging process. This process finds and resolves defects within a program that prevent correct operation of a software. It gives you code documentation due to better coding standards and practices. Unit testing is performed using white-box testing and is either manual or automated.

Integration testing test the connectivity between the couple of unit tested modules. The process of integration testing is the use of dummy programs that are called stubs and drivers. Stubs and drivers don’t look at the entire program logic it just simulates data communication with the calling module. Integration testing is done after unit testing because it gives confidence in performing software integration testing.

System testing evaluate the system’s compliance with the specified requirements. An example of system testing is testing a system that checks sending mail. At this level of testing the complete, integrated system is tested. The reason is to evaluate the systems compliance with the business requirements and assess whether it is acceptable for delivery.

Acceptance means approval or agreement. The user acceptance test is defined as testing the software by the user of client to determine whether it can be accepted or not. This is usually the final testing performed once the functional, system and regression testing are completed. Alpha testing is performed to identify all possible issues like bugs before releasing the product to everyday users or to the public. At this stage you want to simulate real users by using a black box and white box techniques. Beta testing is performed by real users of the software application in a real environment and can be considered as a form of external user acceptance testing. It will be released to a limited number of end-users of the product and the reason for that is to obtain any kind of feedback on the product quality. Beta testing also reduces product failures and provides increased quality of the product through customer validation.

All the different levels of software testing are very important to the quality and reliability of the application. Kapur states “The aim of testing is quality improvement to achieve the necessary reliability. Testing is done at every stage of the software development, in order to verify and validate the software” (Kapur, Singh, Sachdeva, & Tickoo, 2014). Verification is the process of evaluating products of a development phase to find out whether they meet the specified requirements. It’s carried out before validation. The cost of errors caught is less that the errors found in validation. It also explains whether the outputs are according to inputs or not.

Validation is the process of evaluating software at the end of the development process to determine whether software meets the customer expectations and requirements. It’s carried out right after the verification.

Both verification and validation are very important to balance each other out. They both are used to find a defect in different ways. Verification is used to identify the errors in requirement specification and validation is used to find the defects in the implemented software application.

A close up of a map

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA close up of a map

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA close up of a map

Description automatically generated

**References:**

Idrus, H. M., & Ali, N. (2019). Tacit Knowledge in Software Testing: A Systematic

Review. *2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS)*. doi:10.1109/icriis48246.2019.9073275

Kapur, P., Singh, G., Sachdeva, N., & Tickoo, A. (2014). Measuring software testing efficiency

using two-way assessment technique. *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization*. doi:10.1109/icrito.2014.7014679

Tsui, F., Karam, O., & Bernal, B. (2018). [*Essentials of software engineering*](https://ashford.instructure.com/courses/67171/modules/items/3399620)(4th ed.). Jones & Bartlett Learning.