**Week 2 Assignment: Design Modeling Paper**

CST499: Capstone for Computer Software Technology

Amr Elchouemi

James Dickard

03/14/22

Software development is a lengthy, rigorous process that involves an extreme level of detail in the design aspects. Within all the work that goes into the building of a piece of software, a very large amount of that work goes in to testing the program both during development and after. Given that there are a plethora of moving parts in a program that must be tested, it is a given that there are various methods of testing designed to test different aspects of a the system in question. In this paper we will discuss different methods of testing and what those tests are designed to look for as well as how important testing is to the software development life cycle.

Quite possibly one of the lowest levels of testing that is done would be Unit testing, and this level of testing focuses on the smaller parts of the program. Each individual aspect of the program is examined, from individual classes, procedures, or methods, and all these aspects are tested for errors, functionality, and whether the task assigned is completed successfully. This level is quite critical as it ensures that each part of the program, no matter how small, is functioning the way it is supposed to with minimal errors or issues. It is within this level that any inconsistencies or mistakes are discovered and rectified to ensure every component fits into the whole of the design without a hitch.

After Unit testing is completed for all parts of the program that are ready it is time to ensure that all the individual units can work and operate together as a cohesive, comprehensive, and functional unit. These tests are generally required at an early level to ensure that all parts made will work together with the other parts in an effective manner. After this level of testing is completed, it is time to move on to a more in-depth level of testing known as integration testing. Integration testing is the next level of testing beyond functional testing as it takes every single aspect of the system and tests all their interactions with each other as they interact with each other to determine the integrated functionality of the whole system.

Once integration testing is finished the entire system must be tested repeatedly in order to find any additional errors that may have presented themselves with the combination of all the parts of the system. It is common that when one is trying to make two different programs work together there is very often a level of disconnect that occurs between the multiple units at play. This level of testing is designed to search out these bugs and to fix them before they become a real issue. This is generally the final level of testing before Acceptance testing, however when it comes to very large software systems, certain functions can be grouped together into a component, and then multiple components are brought together to form a system. In this situation there arises one additional level of testing that must be done before Acceptance testing. This additional layer of testing is known as Component testing and is done to ensure that each component works both separately and whole. Even though this is an additional layer of testing that is not found in all testing plans, it is extremely common when it comes to large programs or systems with multiple different components, and it becomes ever more crucial the more components that are added to the system.

At this point in the software development life cycle the program is essentially completed and ready to be presented as a final product. However, before the final product can be delivered to the client, there is one final level of testing that must be completed, and this may be the most important level of all. During this final testing time frame the product is presented to the most important people of all, the actual users of the product. The users will then take the program and attempt to use it for the requirements that they set at the beginning of the project. These users will test the program to ensure that it meets and fulfills every single criterion that has been set forward. Only if the program or system passes this final level of testing will it be accepted by the end user and at that point the program is marked as completed.

Throughout the software development life cycle, it has become abundantly clear that testing is admittedly more critical and more important than the actual development of the system itself, for what is the worth of a system that was built but cannot pass any of the required tests? This means that the intensive and rigorous level of testing that is performed is not only relevant but quite possibly the most important part of the software development life cycle itself. Throughout this paper we have explored multiple different levels of testing and determined the importance of testing in general. It is important to remember that all software must not only be designed and programmed properly but must also pass any tests that are required.

A picture containing text, indoor

Description automatically generatedDiagram

Description automatically generated

A picture containing text, indoor

Description automatically generated

Sources

Tsui, F., Karam, O., & Bernal, B. (2018). Essentials of software engineering (4th ed.). Jones & Bartlett Learning.