**Testing in The Software Development Life Cycle**

**CS-230**

Damean Murphy-Short

August 29, 2023

This paper will explore the relationship between testing and software development, specifically as it is concerned as a stage in the software development life cycle (SDLC). In doing so, three subsets of this relationship will be explored: testing as a procedure, the role that testing plays in a successful product, and the points in the SDLC during which testing does or should occur. To begin this examination, it is important that we define a common structure for the SDLC from which we can determine the role that testing plays. This structure consists of various stages that we will take to be in the following order: planning, defining, designing, building, testing, and deployment.

As a procedure, testing seeks to find errors, defects, and failures with the purpose of eliminating them before deployment to a point that is deemed satisfactory relative to the risk associated with defects in the product. This occurs in numerous ways. Some tests aim to find defects in individual components of software products, such as single libraries, functions, or specific pieces of code. These are called “Unit Tests.” Other tests serve to unearth issues that arise when two components are brought together, aptly named “Integration Testing.” Other tests aim to verify the end user experience. All these tests have in common this: they work on a set of requirements that the software should meet to be considered a successful product (the test basis) and examine how well the pieces of code (test objects) conform to this basis by applying them to test cases designed to provide verification of this conformity.

The above also exemplifies the importance that testing holds in software development. It verifies that the requirements set out at the start of production are met. The user tests play an equally vital role in addition, which is the validation that the requirements align with the users’ needs. Without some type of assurance that the product is performing what is required of it, and what has been deemed required is desirable to the user, the product’s success would be up to fate more than preparation. The entire investment of time and energy would be wasted, much like an untested bridge which collapses at the first traveler to cross it.

It is because of this great importance on testing that it takes place within the SDLC, but when should it occur? In our common understanding of the stages laid out at the start of the paper, we see it take place directly before deployment. Surely one does not want to deploy before testing, as we have exemplified in the previous paragraph, but it may also be wise to start testing even sooner. The planning stage should include planning tests, as should the defining stage be the starting point where test bases are set out, as too the design stage should involve designing tests, etc. The earlier one starts testing, the sooner defects are found, and the less time and effort go into removing and redoing work (Hambling, B., et al., 2019). The authors Hambling, et al, would make a strong argument that testing should ideally be involved in the SDLC from start to finish.

References.

Hambling, B., Morgan, P., Samaroo, A., Thompson, G., & Williams, P. (2019). *Software testing : An istqb-bcs certified tester foundation guide - 4th edition*. BCS Learning & Development Limited.