**Southern New Hampshire University**

**Joshua Wozny**

[**joshua.wozny@snhu.edu**](mailto:joshua.wozny@snhu.edu)

**May 1, 2023**

**CS-320: Journal: The Role of Testing in the Software Development Life Cycle**

The Software Development Lifecycle (SDLC) represents the stages involved in implementing a software development project, from ideation to deployment and maintenance. The specific phases vary dependent on the methodology but generally fall into these categories: planning, analysis, design, implementation, testing, deployment, and maintenance (Valacich, J. S., & George, J. F. 2019). There are two primary methodologies for the SDLC, waterfall and agile. Although there are several frameworks within each methodology, they vary primarily in the implementation details.

Testing should be planned as early as the planning phase, as both require extensive analysis to develop properly. During the testing phase, programmers and analysts test individual programs and the entire system to find and correct errors. Testing confirms that all user acceptance criteria have been met and addresses non-functional concerns like security and performance. This phase ensures the quality and reliability of the software and validates that it meets the client’s needs (Valacich, J. S., & George, J. F. 2019).

Testing is handled differently within the agile methodology than in the waterfall methodology. The agile framework develops software iteratively, with each development phase conducted for each iteration. Testing, then, is performed during each iteration and is therefore conducted early and often to ensure that each produced use case functions as desired. The only exception to this rule is integration tests that may be needed to ensure that each of the iteratively produced features works together as intended before the final user acceptance of the application (Charles G. Cobb, 2015).

The waterfall methodology requires a more rigid approach. It iterates through each stage one at a time, with the output of one stage as the input of the next - with little flexibility to return to a previous stage once completed (Cobb 2015). Within this framework, testing occurs near the end of the process as a quality check before user acceptance. Although the waterfall method allows for easier project management and cost estimation, it comes with significant risks. If errors, bugs, or missing functionality are discovered during testing, it will often require starting the project over again from a much earlier stage – both costly and time-consuming (Hambling, B., et al. , 2019).

The V model, an extension of the waterfall methodology, attempts to mitigate these shortcomings by facilitating early work product evaluation.

Diagram

Description automatically generated

**Image from https://media.geeksforgeeks.org/wp-content/uploads/V-Model.png**

In the V model, testing occurs on the work product level so that defects can be identified as early as possible. Nevertheless, each stage must be completed before the next one can be started; this approach to software development still pushes user acceptance and validation to the end of the life cycle, which still creates a risk that the project will need to begin again from an earlier stage – but will generally be less costly or time-consuming than for a standard waterfall project (Hambling, B., et al. , 2019).

REFERENCES

Charles G. Cobb. (2015). The Project Manager’s Guide to Mastering Agile : Principles and Practices for an Adaptive Approach. Wiley.

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.

Valacich, J. S., & George, J. F. (2019). Modern Systems Analysis and Design (9th ed.). Pearson Education (US). https://mbsdirect.vitalsource.com/books/9780135172827