Repeat Agile Software Verification

Sam Cullen

C00250093

South East Technological University
Carlow

Submission Date: 30/08/2024

Sam Cullen – C00250093

Reflection Piece

In this essay, I will discuss the experiences and insights I gained working with Test Driven Development (TDD) and various testing techniques, specifically focusing on unit testing, black box testing, and white box testing while developing the Stay and Gate classes.
After practising the TDD approach, I can confidently say that it gave me a structured and focused way to address the challenges I encountered during this assessment. TDD helped me break down problems into smaller, manageable pieces, making the development process more approachable and manageable.

The red-green-refactor loop, a core aspect of TDD, was particularly valuable. Creating tests, running them, seeing them fail, and then writing the necessary code to make them pass was an iterative process that made me understand the importance of testing early and often. This cycle made debugging easier and served as a roadmap for development. Using TDD, I learned how failing tests could highlight any potential issues early in the process, allowing for more efficient problem-solving and improvement of my code.

In addition to the advantages of debugging and organisation, TDD significantly influenced my design decisions. Writing tests before implementing solutions forced me to carefully examine the structure of my code. This early focus on details led me to create code that was maintainable and scalable. TDD encouraged me to think critically about the long-term effects of my design choices, which led to cleaner, more efficient code. The confidence I gained from using the red-green-refactor loop allowed me to make changes to the codebase with the assurance that any issues I would encounter would be quickly identified and addressed.

Using GitHub to document this process provided a clear and visible record of my development journey. This documentation allowed me to track my progress, reflect on the changes made, and understand how my code evolved. TDD, combined with version control, became a powerful tool that strengthened my development skills and provided me with a framework for continuous improvement.

However, TDD also came with challenges. Getting used to the red-green-refactor loop was initially difficult. The mindset of intentionally writing failing tests before the actual code felt inconsistent at first, and changing to this new approach took time and persistence. Writing these tests took a lot of time at first, which seemed like a burden and slowed down the development process.

Despite these initial setbacks, I found that as I became more comfortable with TDD, the benefits began to outweigh the challenges. Writing failing tests and then the code to make them pass became natural. Detecting and fixing bugs early in the development process, rather than at the end, turned out to be incredibly valuable. Although it took time to fully grasp the principles of TDD, I now believe that the long-term benefits, in terms of code quality, bug prevention, and maintainability, made it a worthwhile investment.

Looking back, the principles I learned from using TDD have fundamentally changed how I approach coding. The combination of TDD with black-box and white-box testing techniques has given me a more comprehensive toolkit for ensuring code quality. Black box testing taught me to think from the user's perspective, focusing on the input-output relationship without worrying about the internal workings of the code. White box testing, on the other hand, helped me achieve 100% branch coverage, ensuring that every possible path in the code was tested.

I've learnt that TDD and rigorous testing can have a significant impact on the reliability and quality of my work as I continue to build and test code throughout my career.
The skills and insights I gained through this process will be invaluable as I continue to grow in my career as a software developer. This experience has not only strengthened my confidence in using these techniques but has also given me a solid foundation for taking on future projects with a more disciplined approach.