**Final Project**

**Test Driven Development**

**Jeremy Pritchard Student Lakeland Community College**

**Professor Grasser ITON 1011**

1. **Introduction**

Software engineering over the years has evolved in many ways to have better ways to program and to develop software in general. One of the ways that has evolved is called Test Driven Development where in essence the test is written first and then the real code is written after the test fails for the first time. The general reason for doing this is so that code comes out less bug ridden. However, we will go into a more in-depth analysis of where test driven development came from and how it works.

1. **What is Test Driven Development?**

Test driven development is the process of writing a test case for a software requirement, then writing the actual code for the requirement once the test fails initially. See the side figure from Scott Ambler [2] for an example of how the steps of test driven development work. After the test pass you write a new test that’s meant to test the next small feature you are going to add to the code. If the test passes it means you need to write another new test. However, if it fails you make small changes to the actual code until the code passes, at which point you write a new test. You also keep all the tests and when you run one test you run them all to make sure that you are not breaking something that a previous test made sure was working properly. The last step you do that the picture by Ambler is that after the test passes you are to refactor your code. Refactoring your code is by removing duplicate logic from your code [3]. When you refractor your code you still will want to run the tests to make sure you don’t remove or change the way your code works. For example, if you are expecting the output of your function to be 2 you do not want to change your code in a way that know your output is 5.

Figure 1 Steps of Test Driven Development [2]

* 1. **History Test Driven Development**

Test driven development(TDD) in some form was start back in the 1960’s with mainframe programming [6]. One way a programmer could verify if their code was correct before running it through the mainframe was to write themselves the expected outcome of the code. This way the could verify where exactly their code went wrong when they received their output. Kent Beck in the early 2000’s is attributed with making TDD well known with his book *eXtreme Programming* and *Test-Driven Development by Example* [2]. Since then it has been incorporated into Agile Programming and has been widely adopted in some form in programming.

* 1. **Test-driven development example**

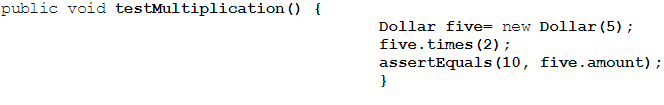
To show how TTD works you will first want to write a test that is going to test the code you are about to write. As the example picture form Beck [3] shows you have written a test to test code that you are about to write This test should fail the first time since no code should be written that allows the test to pass yet.

Figure 2 A Sample Test [3]

1. **Would I use it?**

Would I use test driven development, yes I think I would use test driven development in some form. I don’t think I would use it in its purest form as it seems a bit to slow paced for my style of coding. However, I feel writing some tests and then writing the main functions/method/classes based off those tests could help you write better code and have less bugs remaining in the code. I don’t think it would use it in writing all code as some code is just obvious as to how it needs to be written. With how the process is that could be a lot of wasted time on testing something that is either to simple or obvious to need extensive testing. I would also be hesitant to use it if I don’t fully understand the requirements of the project yet. As some projects may have changing requirements or not fully developed requirements, with TDD you need to have a complete and full understand of the requirements [5]. So, without having the complete understanding it could be that you could create a test and then code that could lead the design in the wrong direction.

* 1. **Have I ever used test driven development?**

I have not used test driven development in my own programming yet. I would say that I have used parts of the philosophy in my programming to test and ensure that my code is working properly. For example, I will typically write smaller amounts of code and test those parts before moving on to the next part. I also will to some degree refactor my code to make it simpler to understand, but also testing it to make sure nothing changes the function of my code.

* 1. **How would I use test driven development?**

How I would use test driven development would be at work as I had to write or modify any kind of python or visual basic scripts for sites. I would use it to test any method or function that I had to change to make sure what I did worked correctly. I would also use it in python scripting while working on any personal projects that I had at the time.

1. **Final Thoughts**

Test driven development is a step forward over the past where such practices weren’t around or lost due to the constant change of the industry [6]. It is a tool if used correctly could help to reduce the amount of errors that are in code, though may not help solve issues with integration of all the components [1]. It should be something that each developer at least utilizes in some extent whenever writing code.

* 1. **After learning about it would I follow it?**

After all the research into yes, I would follow TDD at least in the beginning when starting to create the design of the project. The automatic unit testing and any of the JUnit type testing tools are invaluable resources to use in the beginning of the project as well [5]. However, as time went on in the project I feel with integrating multiple components of the project together it may not be as useful of a tool to use.

**References**

[1] “Test-driven Development,” Wikipedia, 18-Nov-2017. [Online]. Available: https://en.wikipedia.org/wiki/Test-driven\_development. [Accessed: 04-Dec-2017].

[2] S. W. Ambler, “Introduction to Test Driven Development,” Agile Data, 2013. [Online]. Available: http://agiledata.org/essays/tdd.html. [Accessed: 04-Dec-2017].

[3] K. Beck, Test-Driven Development by Example. Boston, MA: Pearson Eduction Inc, 2003.

[4] R. Champman, N. White, and J. Woodcock, “What Can Agile Methods Bring to High-Integrity Software Development?,” Communications of the ACM, vol. 60, no. 10, pp. 38–41, Oct. 2017.

[5] “Test Driven Development (TDD),” Agile Testing Framework. [Online]. Available: https://www.agiletestingframework.com/atf/testing/test-driven-development-tdd/. [Accessed: 04-Dec-2017].

[6] D. Barber, “Why Test-driven Development?,” Derek Barber, 27-Mar-2012. http://derekbarber.ca/blog/2012/03/27/why-test-driven-development/.