FIT 4004 – System Validation and Verification

Assignment 3

Test Analysis Report

Demetrios Christou

Introduction

This report is part of Assignment 3 detailing the test process for the project.

What bugs, if any, did you find with your unit tests?

After the code for parsing the time zone in the argument parser was written, I wrote some extra tests and one test that failed was the -00:00 time zone and this was caused by an if condition only checking if the variable was less than zero and was fixed to check if its less or equal to zero.

Other than bugs the tests helped when some refactoring was done on how the date time was handled and the tests made sure that the refactor did not create any additional bugs – The refactored code was tweaked until the tests passed.

Do you think your testing strategy resulted in adequate tests?

Since our testing strategy was to test for functional correctness and boundary cases, I think it resulted in a reasonable number of tests, however more edge cases may remain we did not think about, or tests that we have not considered relate to the functionality.
We did split main functionality into separate modules witch did make writing the tests a lot easier and the amount required a lot less, it also helped with avoiding duplicate tests between the modules, for example the argument parser made sure the date time variable is the right type and format so the other modules need not to test it because they get it from the argument parser class.

Do you think it works better for the programmer responsible for code to write the unit tests for it, or for somebody else to

do it?

I think this goes both ways. In one hand, the one that wrote the code is more familiar with it and may find it easier to write the tests, however because he wrote the code he may be a bit bias on his tests and is less motivated to find bugs or he may not consider some due to the knowledge on how the code works. On the other hand, a programmer that did not write the code, will have a different perspective and test purely on the specification functionality and boundary cases without any bias on how the code was written. Both have its pros and cons, and I think a combination of both would work best.
I think the most important factor is the familiarity with the specification and the experience in writing tests.

How long did writing mock code take? Was it a major component of the overall effort?

Most of the time was spend trying to understand how the mock code works and how to patch the API, after understanding the mock code was not that complicated in our project since the only parts we needed to connect to the network was when fetching the tweets, and when posting the graph.

What Would I do Differently Next Time

I think next time I would focus more on writing the main tests for boundary cases and functionality before writing the functional code and then write the code with the aim to make the tests pass.
Also having some more random/edge case tests could help reveal more bugs.