Test Approach

This document examines different possible testing approaches that can be applied to the Pizza Dronz project, and justifies which approaches were ultimately chosen.

Testing approaches

Proactive testing

Proactive testing involves creating the tests during the early phases of the project and basing them solely off of the requirements. The tests used here called black box tests, and they just verify the specification so do not need any information about the current implementation of the system such as Unit tests.

Reactive testing

Reactive testing is perhaps the opposite of proactive testing and involves creating the tests once the software is complete. As the analysis takes place once the system is complete all the information about the implementation is known and so the tests that will be used will likely be white box tests which are more concerned about the structure of the code.

Combination

A combination of both proactive and reactive testing can be implemented to attempt to get the benefits from both strategies. This would involve implementing proactive tests such as unit tests at the beginning of development based on the specification and then adding grey box (tests in which a limited amount of the implementation is known) and white box testing if and when any issues do arise. This combination can allow the testers to catch problems while they are coding whilst also making sure that the code can be readily tested as they go.

Iterative testing

Iterative testing is slightly like the combination method, except that testing would be carried out at predefined moments in development. The coding process would be divided into sections or iterations and at the end of each section testing would occur, any bugs would be corrected and then development would continue. This would again use black, grey and white box testing and would make for a very structured testing approach.

Continuous feedback and automated testing

The continuous feedback and automated testing approach by DevOps involves setting up a pipeline that will automatically test any new code that is written and generate the appropriate feedback immediately whenever a bug is encountered. This is a very useful

and sophisticated method but only makes sense to implement on a much larger project than the Pizza Dronz.

Testing approach choice

Taking into account effectiveness of these methods, the timing of development and the potential resource of the choices, using the combination approach is the best suit for the project. A combination of proactive and reactive tests would allow for us to adapt our testing as it seems suitable as the development process goes on. It will allow us to ensure that all manor of requirements are met suitably and correctly before the system is released. Whist I will be implementing Black, grey and white box tests as the development continues such as Unit tests using Junit, it is also important to continuously manually inspect my code in an attempt to spot errors or areas that may need testing. This approach will allow me to test the code thoroughly whilst not having to spend an absurd amount of time or resources to achieve this.