

# Software Testing Portfolio

## Outline of the Software Being Tested

This portfolio is about testing performed on the ILP Pizza Dronz project. This system was a pizza delivery service that used drones to collect pizzas from local restaurants and deliver them to the top of Appleton tower. The software that I developed handled order validation, and the calculation of flightpaths for the drone which was then stored in three separate formats. The specifications for this project can be found in the ILP coursework specifications [1]. All additional supporting documents can be found in the GitHub repository which will be linked at the end of this portfolio and here: <https://github.com/Gregory12321/SftwrTstg>

## Learning outcomes

### 1. Analyse requirements to determine appropriate testing strategies [25%]

#### 1.1. Range of requirements, functional requirements, measurable quality attributes, qualitative requirements, ...

I analysed potential stakeholders for the system, and discussed what requirements they would have for the system. I list both functional and non-functional requirements of the system, as well as some measurable quality attributes of the project. This can be found in System Requirements [2].

#### 1.2. Level of requirements, system, integration, unit.

I identified which of the three levels of requirements would be adequate in testing each requirement that I had previously identified, and justified why this would be the case. This can be found in Requirement Levels [3].

#### 1.3. Identifying test approach for chosen attributes.

I evaluated different possible testing approaches and their potential benefits. This can be found in Test Approach [4].

#### 1.4. Assess the appropriateness of your chosen testing approach.

After evaluating the different testing approaches I decided on the approach that I would undertake and explained why it would be beneficial for this project. This can be found in Test Approach [4].

### 2. Design and implement comprehensive test plans with instrumented code [25%]

#### 2.1. Construction of the test plan

Three different level requirements were examined to show the understanding of how to create a comprehensive test plan. These were prioritised, and a testing strategy was made for each. This can be found in Test Plan [5].

#### 2.2. Evaluation of the quality of the test plan

For the three different requirements, the risks involved with these requirements and with the testing of them were outlined, and how much time should be scheduled to deal with each of them. This can be found in Test Plan [5].

### 2.3. Instrumentation of the code

The scaffolding and instrumentation needed for each of the three requirements were listed and then explained. This also considers how much time the scaffolding may take to develop and therefore how early on in development they should be scheduled for. This can be found in Test Plan [5].

### 2.4. Evaluation of the instrumentation

The scaffolding and instrumentation needed for each of the three requirements were listed with an evaluation of what could be improved given a different timeframe and how well the current instrumentation will work. This can be found in Test Plan [5].

## 3. Apply a wide variety of testing techniques and compute test coverage and yield according to a variety of criteria [default 20%]

### 3.1. Range of techniques

Three different level requirements were examined to demonstrate a range of testing techniques that would be used throughout the testing process. This can be found in Test Implementation [6].

### 3.2. Evaluation criteria for the adequacy of the testing

The adequacy criteria was explained for the three different level requirements, including how much coverage was expected for different priority requirements. This can be found in Test Implementation [6].

### 3.3. Results of testing

The results of the planned testing of these three requirements were presented and summarised, and any issues were discussed and explained. This can be found in Test Implementation [6].

### 3.4. Evaluation of the results

These results of the testing of these three requirements were evaluated, and any improvements that could be made to the testing were given. This can be found in Test Implementation [6].

## 4. Evaluate the limitations of a given testing process, using statistical methods where appropriate, and summarise outcomes. [15%]

### 4.1. Identifying gaps and omissions in the testing process

Any gaps in the testing process were discussed, and ways to improve the testing to eliminate more of these omissions were proposed. This can be found in Test Evaluation [7].

### 4.2. Identifying target coverage/performance levels for the different testing procedures

The different targets of coverage across the system were stated and explained. Four different coverage metrics were used to judge this, method coverage, branch coverage, pass rate and requirement coverage. This can be found in Test Evaluation [7].

#### 4.3. Discussing how the testing carried out compares with the target levels

The results of the coverage testing across the system were stated and were compared to the previous targets set out, in all four previously mentioned metrics. This can be found in Test Evaluation [7].

#### 4.4. Discussion of what would be necessary to achieve the target levels.

The results of the coverage testing across the system were stated and were compared to the previous targets set out, in all four previously mentioned metrics. This can be found in Test Evaluation [7].

### 5. Conduct reviews, inspections, and design and implement automated testing processes. [15%]

#### 5.1. Identify and apply review criteria to selected parts of the code and identify issues in the code.

Various methods of code review approaches were used and explained, and then the resulting changes made because of these reviews were stated. This can be found in Reviews, Inspections and Automation [8].

#### 5.2. Construct an appropriate CI pipeline for the software

The construction process of a CI pipeline was explained, as well as how it would be constructed for this project. This can be found in Reviews, Inspections and Automation [8].

#### 5.3. Automate some aspects of the testing

Different ways in which automation could potentially be implemented into the testing process were mentioned, as well as their potential benefits and deficits. This can be found in Reviews, Inspections and Automation [8].

#### 5.4. Demonstrate the CI pipeline functions as expected

The ways in which the pipeline will function and improve the development process were explained, as well as explaining how the feedback loop it creates is very beneficial to programmers. This can be found in Reviews, Inspections and Automation [8].

GitHub: <https://github.com/Gregory12321/SftwrTstg>