Software Testing 2023-4: Your Portfolio

[Text in grey is guidance and should be deleted in your final portfolio. Text in blue is the statement of the learning outcomes - you should retain this. The text of your portfolio should be in black. To structure your self-evaluation, you should retain the statement of the sub-criteria for each learning outcome. Your text should evidence how well you have done is reaching the learning outcome. ]

The aim of the portfolio is to provide evidence that you have taken the elements of the learning outcomes into account and have done some work that demonstrates you have achieved the learning outcome to some extent. The idea here is to point to the work you have done that evidences your having achieved the learning outcome. You should try to be as specific as possible, pointing out how particular pieces of work illustrate your learning. Notice that you are not expected to produce a “perfect” answer. It is important to remember that you have a time budget and you need to manage that carefully to produce a balanced portfolio.

For each of the learning outcomes a quiz will be available that tests basic knowledge. You can choose to use your performance in the quiz as evidence of knowledge or not as you choose (provided you have passed the quiz for a particular Learning Outcome you should not mark yourself below 2 on any sub-criterion). The quizzes are not compulsory and if you choose not to use them as evidence they will not be included in the assessment.

Outline of the Software Being Tested

This section should have a brief overview of the software being tested. It should also have a link to the repo with your work on the software in a form that is readable by the auditors. If your chosen software is confidential, you do not need to make source code available.

PizzaDronz offers a solution to students needing food in a timely manner. The system will implement an algorithm to control the flight of the drone as it makes its deliveries while respecting the constraints on drone movement. The system will integrate with an online system developed by the School of Informatics, that will track the placement of pizza orders. By addressing the nuanced process of pizza ordering and delivery, PizzaDronz enhances the overall experience and well-being of students. [Link to work]

Learning Outcomes

This section demonstrated how well you have reached the learning outcomes by considering each sub-criterion.

1. Analyze requirements to determine appropriate testing strategies [default 20%]

* 1. Range of requirements, functional requirements, measurable quality attributes, qualitative requirements, …

The evidence to support this criterion is a list structured by types of requirement that good coverage of different types of requirement. This should be a short document stored in your GitLab project that this section of the portfolio can refer to.

**Functional Requirements:**

* The system must fetch orders from the REST service (the database storing order information). Orders should include information about the restaurants, the menus, and the customer's information.
* The system must process and validate each order to ensure its correctness and completeness. Details such as order number, order status, and the order validation code should be extracted[1].
* The system should determine the optimal movement of the drone from a restaurant to the drop-off point by considering the location of restaurants, the drop-off point and the central area whilst avoiding no-fly zones[1]. It must calculate the flight path for each move made by the drone, recording relevant details such as the flight path of the drone. The planning and plotting of the flightpath should have a runtime of 60 seconds or less[1] to minimise delays in the delivery of pizzas.
* The system must generate three types of files. These include deliveries in JSON format, flight paths in JSON format, and a GeoJSON file for visualization.

**Non-Functional Requirements:**

* The system must have robust mechanisms for failure reporting, particularly in the event of a drone crash or system malfunction. Real-time monitoring and alerting systems should be in place to detect anomalies or unexpected behaviours during drone operations such as the wrong flight path being taken.
* The system must have user privacy and anonymity to uphold ethical standards and legal compliance. The system must ensure the privacy and confidentiality of customer information and order details. The system must ensure that the drone's flightpath and flightpath data does not compromise the privacy of residents.
* The system must be able to effortlessly handle a substantial volume of daily orders, ensuring swift order retrieval from the REST service and efficient drone flight path calculations.
  1. Level of requirements, system, integration, unit.
  2. Identifying test approach for chosen attributes.

Chosen attribute is 60 second runtime test

1.4. Assess the appropriateness of your chosen testing approach.

1. Design and implement comprehensive test plans with instrumented code [default 20%] 2.1. Construction of the test plan

2.2. Evaluation of the quality of the test plan

2.3. Instrumentation of the code

2.4. Evaluation of the instrumentation

* To support your portfolio section on LO2 you should construct a test planning document. This provides the evidence you need to refer to in the LO 2 section.
* This is the most challenging LO to assess fully because a fully complete answer could demand an account of the process you intend to use.
* To make this manageable, the test planning document should outline the factors constraining when and how a small number of your requirements are to be analysed and tested. AND
* Should choose one of the lifecycle approaches outlined on the Y&P Chapter 20 slides and discuss where in the lifecycle those requirements could be tested.
* Should consider any risks arising in your chosen mapping of a requirement to the process

1. 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

3.2. Evaluation criteria for the adequacy of the testing

3.3. Results of testing

3.4. Evaluation of the results

1. Evaluate the limitations of a given testing process, using statistical methods where appropriate, and summarise outcomes. [default 20%]

4.1. Identifying gaps and omissions in the testing process

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

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

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

1. Conduct reviews, inspections, and design and implement automated testing processes. [default 20%]

5.1. Identify and apply review criteria to selected parts of the code and identify issues in the code. [default 20%]

5.2. Construct an appropriate CI pipeline for the software

5.3. Automate some aspects of the testing

5.4. Demonstrate the CI pipeline functions as expected.