Test Plan

- **Introduction**

  The Test Plan outlines the methodology and test cases planned for the PizzaDronz project. The project adopts a Test-Driven Development approach to ensure that the project satisfies the verification criteria.

- **Scope**

  The test plan outlines the testing strategy and test sets planned for the PizzaDronz project. The project employs a test-driven design approach to ensure that it meets the specified requirements. The tests will focus on the software requirements outlined in the requirements list. The hardware requirements, which involve the physical components of the drone, will not be included in this test plan, as they will be the responsibility of the hardware team to test. User testing will be developed at a later stage.

- **A section discussing the priority you put on the requirements and what they need in order to be adequately tested.**

  The requirements list outlines the majority of the specifications that the system must adhere to. The functional requirements are of utmost importance and must be met in order to ensure that the system has been fully developed according to the given specifications. Within the functional requirements, the "Safety" section is of paramount importance, as it aims to minimize harm during system operation and drone delivery. The "Correctness" section ensures that the drone can carry out deliveries without errors and that consumers are able to receive their orders.

  Measurable quality attributes are quantifiable characteristics of the software system that can be evaluated and measured. These attributes are used to assess the overall quality of the PizzaDronz system and to identify areas that require improvement. All sections within the measurable quality attributes should be given equal priority, as they are all necessary in evaluating the system's performance.

  The qualitative requirements, although subjective and difficult to measure quantitatively, define the overall quality and user experience of the PizzaDronz system. These requirements help the system to meet validation needs. Although the priority of these requirements should be lower, a well-designed system should still meet these requirements in order to provide a better user experience.

- **A section describing the scaffolding and instrumentation that is necessary in order to analyse and test adequately.**

  The concept of scaffolding in software testing refers to the creation of a simulation environment that mirrors the actual production environment. The purpose of scaffolding is to provide a controlled and realistic testing environment that enables the testing team to identify and rectify potential issues before the product is released into the market.

Scaffolding can also be used as a placeholder for missing parts of the system configuration to allow for the testing of functionalities.

Instrumentation involves the insertion of code into the system to collect data about its behavior during testing. The goal of instrumentation is to provide additional testing strategies that help monitor system performance and functionality. For instance, by incorporating a timer into the system test, one can verify that the time required to generate the three output files is within 60 seconds (as specified in the measurable quality attributes). Instrumentation helps to continuously evaluate system performance by tracking the time taken during subsequent development phases.

- **A section demonstrating you can work out where to put a testing activity into a lifecycle, so you have the capacity to develop a comprehensive test plan.**

The DevOps is a life cycle that used to build, test and deploy the software.

Create: The system is developed under version control by using GitHub. This phrase involves determining project goals and objectives and generate general project plan and test plan.

Verify: This step involves testing and validating the code and application to ensure the project has satisfy most quality and performance standards. Most of the functionality test should be pass during this step.

Package: This step involves packaging and preparing the code and applications for deployment. The security test and other tests must be pass.

Release: This step involves deploying the code and application to the production environment.

Configure: This step involves configuring the production environment and re-run all the test to ensure the code and product are set up correctly and able to run.

Monitor: This step involves continuously monitoring the code and applications to ensure they are still functioning as expected and all the tests are still passed.

Plan: This step involves planning and organising future development.

- **Evaluation of the quality of the test plan**

The test plan should delve into the details by offering a scaffolding practical example from the testing process. The adoption of the DevOps lifecycle is a logical choice, but further elaboration would be beneficial. In general, the test plan for the PizzaDronz project outlines the use of a test-driven approach to guarantee that the final product satisfies its verification requirements. The scope section of the plan explicitly states that the tests outlined in the document are only used to validate software requirements.