Test Plan

- Introduction

  The test plan outlined the test approach and test sets planed for the PizzaDronz project. The project uses test-driven design approach to ensure the project meets the verification requirements.

- Scope

  The tests should have covered most software requirements that are mentioned in the requirements list. The hardware requirements test involves actual components of drone will not be included in this test plan, as it is responsible to be tested by the hardware team.

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

  The requirement list outlined majority of the requirements that system should have follow. The functional requirements must be meet to ensure the system is fully developed by the given specification. Hence, it should have highest priority among other requirements.
  In functional requirements, the section "Safety" are most important requirements that minimize the harm during system operation and drone delivery. The 'Correctness' section ensures drone can complete the deliveries without causing any errors, and consumers are able to received their deliveries.

  The measurable quality attributes are specific characteristics of a software system that can be quantitatively measured and evaluated. These attributes are used to assess the overall quality of the pizzaDronz system and to identify areas that need improvement. The priority of each section should be same as they are all needed to evaluate the system performance.

  The qualitative requirements are subjective and difficult to measure quantitatively, it is used to define the overall quality and user experience of the pizzaDronz system. These requirements help the system to meet the validation needs. The priority of these requirements should be least, but a good system design should have meets them to allow better user experience.

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

  The scaffolding refers to the process of a creating a test environment that simulates the condition of actual production environment. The goal of using scaffolding is to create a realistic and controlled testing environment that allows test team to modify and fix any potential issues before release the product to the market. The scaffolding can used as place holder for any unwritten parts to pass the system configuration, so the functions can be tested.

  Instrumentation refers to the process if adding code to the system, so the data regarding

system behaviour can be collected during test. The goal of using instrumentation is provide additional test strategy to helps monitor system performance and functionality. For example: by adding timer in the system test, it ensures the time for system generating the required 3 output files is within 60 seconds (measurable quality attributes). And it helps to further evaluate the system performance by keep track of the time costed during further development.

- 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.