System Test Plan

For

Turtle Tech

Team members: Christopher Angland, Laurel Dodson, Parker Brown, RoseEllen Hoke,

Version/Author	Date
Version 1	19 October 2022
Version 2	8 December 2022
Version 3	2 March 2023

Table of Contents

1. Introduction

1.1 Purpose

This document is a test plan for TurtleTech System Testing, produced by the System Testing team. It describes the testing strategy and approach to testing the team will use to verify that the application meets the established requirements of the business prior to release.

The purpose of the TurtleTech project is to produce a hot swappable payload casing for the Jetson Nano and streamline the process of learning and setting up the neural network system on the Nvidia Jetson device. The payload casing will be 3D printed and tested for durability and conservation of space. The TurtleTech team will produce documentation on how to go from having a Nvidia Jetson in a box to powering it on and running the neural network on the Jetson Nano. The TurtleTech team will also assist at the flights in order to learn how the neural network functions and how to debug any errors that occur during the flight.

1.2 Objectives

- Meets the requirements, specifications and the Business rules.
- Supports the intended business functions and achieves the required standards.
- Satisfies the Entrance Criteria for User Acceptance Testing.
- The payload case will be able swappable to other drones
- The payload case will be tested for durability and longevity
- The payload case will conserve as much space as possible
- The team will learn how to get the neural network functioning on the Nvidia Jetson and write documentation on how to get the Jetson running
- The team will research and learn how to adapt the turtle AI to whales

2. Functional Scope

The Modules in the scope of testing for the Turtle Tech System Testing are mentioned in the list below:

- Attachment to chassis
- Durability of payload case
- Construction of Payload
- Ease of Assembly
- Portability of Whale AI
- Neural Network Reconfiguration
- Speed of Deployment

3. Overall Strategy and Approach

3.1 Testing Strategy

Turtle Tech System Testing will include testing of all functionalities that are in the defined functional scope of section 2. To do so, these metrics must be evaluated independently, as they affect different aspects of the system's ability to move between different housings and various targets. For the physical payload model, initially most of the case testing itself will be in virtual fusion simulations. The team can print a smaller model to check the plastic strength for a physical payload test, and then a full-size model later on. It's important to note that the final test would be accomplished via sending out the payload on the UAV itself for a full check of functionality. The whale AI would be evaluated on how well it runs on a PC, with reference statistics of how well it evaluates whale images on the original contest, and if possible, tests would also be done on our own images to find how accurately the AI finds whales. Of course, the final challenge of the Whale AI would be showing it running on the Jetson board, with further testing of its accuracy becoming secondary at that point, as the operation onboard the hardware is the first, and most vital, priority.

3.2 System Testing Entrance Criteria

Testing of both sides of the system can begin once a 3D model of a potential payload case has been created, and for the software, once an executable version of the correct Whale AI is obtained. The initial 3D model test can take place virtually, via the potential ability to model stress tests through Autodesk's Fusion 3D modeling software. Similarly, an executable version of the Whale AI should be able to take in photos and output processed images, in which case those images can be entered into TensorFlow-based systems to begin working towards an AI setup that would conceivably port to the Jetson. Thus, just a PC-based executable of the Whale AI is all that is necessary for initial testing of the software system, and initial hardware testing can be accomplished digitally as well. To finish some physical tests, however, a physical model of the payload case would need to be created, most likely via 3D printing, so that's a barrier criterion to some of the intended tests for now.

3.3 Testing Types

3.3.1 Usability Testing

Usability Testing will involve checking the software for ability to run on the PC at first, and then evaluating it for accuracy of identification (this time for whales). Following that, the ability of the system to port and run on the Jetson hardware will be paramount; it is unlikely that the accuracy of the system could be easily evaluated following that, so merely running on the actual Jetson hardware is the intended test. The simplest methodology for usability testing is post stress-test simulation, where a physical model could be checked for how well it fits into the current UAV chassis, how the components such as the Jetson fit into the model, how quickly it can be loaded into the UAV, and whether any additional features, such as a switch, works for the system. In payload testing the focus will be on ease of use and application; in software the main aspect is proof-of-concept, whether or not it can be ported and run on the current hardware.

3.3.2 Functional Testing

This project requires a processing system that collects photos of turtles via case-mounted cameras and a circuit board which are positioned in a UAV. It also requires an operating AI to scan through the images for identifying the wanted information. The business requirements are thus having a functioning payload to capture and store images, having the software to process them, and having the physical containment to keep the entire setup within the small area allotted to the purpose.

Again, the images are not private, and business requirements are less profit-minded, so typical business rules and conditions for a for-profit endeavor are not a priority for this test. The test-bank of photos being provided to the Turtle Tech team is not private, and currently the product is not being presented to the customer. In the future, the main rule is that the product meets the functional requirements.

3.4 Suspension Criteria and Resumption Requirements

This section covers the reasoning and requirements of circumstances to suspend testing of the system or a portion of it, and the necessary steps that must be taken prior in order to resume testing in such an occurrence.

3.4.1 Suspension Criteria

Testing will be suspended if the hardware team brings up a concern that must be addressed in the new design of the payload case. This is why it is important to keep in communication with other sects of the larger team; all aspects of functionality must be maintained. In the same vein, if the proposed design interferes with the running of other, previously established systems, which would be uncovered during, testing will then need to be suspended as well. Regarding the Whale AI, testing will be suspended until further information can be gathered regarding the intended AI to apply to this system and situation, as additional background knowledge is required at this point so as not to select a program that could disrupt the current system setup.

3.4.2 Resumption Requirements

Once the concerns of other interested parties are addressed, the system can resume testing. Any issues caused by the modifications to the payload case would need to be resolved prior to resumption, to eliminate problems of interfering with the other hardware. In addition, the design must ensure that functionality of the cameras and overall weight distribution is maintained, as otherwise the new payload case could negatively impact the work of other sections of the larger Turtle Tech team. Additionally, if information is gathered regarding the Whale AI which may means the system must be revised (to a new software setup, or some other alteration as suggested by those who have previously researched the Right Whale problem) then after that adjustment is made, testing can be readjusted and then resumed.

4. Execution Plan

4.1 Execution Plan

The execution plan will detail the test cases to be executed. The Execution plan will be put together to ensure that all the requirements are covered. The execution plan will be designed to accommodate some changes if necessary, if testing is incomplete on any day. All the test cases of the projects under test in this release are arranged in a logical order depending upon their inter dependency.

Requirement (From SRS)	Test Case Identifier	Input	Expected behavior	Pass / Fail
3.3 Verify script functionality on local machine	1.1	Shell script	Test script on a folder, iterating through each item	TBA
3.3 Functionality of test environment	1.2	Nvidia Jetson Nano	Power on successful	Pass
Verify mounting system secures designated components	1.3	Mount for designated components	Rigid connection between the mount and baseplate of the payload	TBA

Verify switch and LED indicators properly control and indicate the flow of power in payload	1.4	User input of button/switch activation	The respective cutoff and flow of power from the battery, LED indicators turn on for Jetson payload	TBA

5. Traceability Matrix & Defect Tracking - Chris

5.1 Traceability Matrix

List of requirements, corresponding test cases

Requirement CRITICAL: System requirements Specification, ?.?.?: The shell script shall iterate through each item on a local machine

Test Cases: The script operates on a local machine

Requirement CRITICAL: System requirements Specification, ?.?.?: The Nvidia Jetson shall run on the test environment

Test Cases: The Nvidia Jetson powers on

Requirement CRITICAL: System requirements Specification, ?.?.?: The mounting system shall be sturdy when locked into place

Test Cases: Mounting system does not shake in the drone

Requirement CRITICAL: System requirements Specification, ?.?.?: The LEDs shall signify when the paylaod is powered on

Test Cases: The LEDs light up when the payload is powered

5.2 Defect Severity Definitions

Critical	The defect causes a catastrophic or severe error that results in major problems and the functionality rendered is unavailable to the user. A manual procedure cannot be either implemented or a high effort is required to remedy the defect. Examples of a critical defect are as follows: • System abends • Data cannot flow through a business function/lifecycle • Data is corrupted or cannot post to the database
Medium	The defect does not seriously impair system function can be categorized as a medium Defect. A manual procedure requiring medium effort can be implemented to remedy the defect. Examples of a medium defect are as follows: • Form navigation is incorrect • Field labels are not consistent with global terminology
Low	The defect is cosmetic or has little to no impact on system functionality. A manual procedure requiring low effort can be implemented to remedy the defect. Examples of a low defect are as follows: • Repositioning of fields on screens • Text font on reports is incorrect

6. Environment

6.1 Environment

The payload casing will go through multiple different types of environment testing. The payload case will go through a drop a test, temperature testing, and a mechanical test. Once the payload case has passed all three of these tests the Jetson Nano will be installed, and the payload case will be given to the hardware team for further critiques and testing.

7. Assumptions

This test plan for an image based neural network requires base assumptions in order to more accurately define a successful test. For an isolated software test, all assumptions are based around the external inputs and outputs available to the software being correct for the true application of the project.

The inputs are the power for the computing system, and the images provided to the software. An obvious requirement for software to run is having power for the processor the software is being executed on. This has been an issue in the past, however solutions for this problem are being implemented in the real application of the project so it shouldn't be an issue. The images being provided to the software for this test are from a training set specifically for this project. It is to be assumed that these images that are being inputted to the neural network are of the same resolution that the real project will be receiving, and the contents of the image are the correct target that the software must identify.

An output for the system being tested also must be installed properly so that the capability to measure the success of the software exists. Currently, the planned output for the system is storing saved images onto an SD card. Without this storage device, the system would not have the allocated storage to save the output of the software.

8. Risks and Contingencies

The hot swappable payload brings in the aspect of some risks when testing the case. The hardware test can cause some safety hazards, equipment damage and schedule delays. In order to help mitigate these risks the TurtleTech team will always operate with safety at the forefront, and in conjunction with the TurtleTech hardware Team. All aspects of the design before printing and testing will be discussed with team members of both this group and the hardware team. Any modifications made to the payload will be done with previous confirmation of the hardware team to make the specified modifications.

Small-scale prototypes shall be considered first when testing any mechanical systems before attempting full-scale testing. For electrical systems, critical hardware, (any device capable of computing), shall not be connected to a system while testing. Critical hardware shall only be connected to a system once its' safe functionality has been verified. In case of any failure or emergency, the proper procedures for said emergency as instated by the Micaplex Lab shall be followed.

9. Appendices