**Software Test Plan**

NASA Maestro EVA/IVA Heads-Up Display (HUD)

Version 4.0

April 26, 2020

*Prepared by Ali Alnaqeeb and Laura Tamshen*

**Team Members**

Ali Alnaqeeb

Laura Tamshen

Ryan Cohan

Matthew Elliott

Jonathan Johnson

**Index of Revisions**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Name | Date | Description |
| 1.0 | Ali Alnaqeeb | 03/02/2020 | Initial draft and structure set up |
| 1.1 | Laura Tamshen | 3/10/2020 | Update test cases to include comparison to baseline document |
| 2.0 | Ali Alnaqeeb | 03/21/2020 | Updated document to correspond with all project documentations |
| 3.0 | Ryan Cohan | 04/05/2020 | Update document formatting |
| 4.0 | Ali Alnaqeeb | 04/24/2020 | Updated document with negative test cases |

**TABLE OF CONTENTS**

[1. Introduction 5](#_Toc38559050)

[1.1. Purpose 5](#_Toc38559051)

[1.2. Scope 5](#_Toc38559052)

[1.2.1. In-Scope 5](#_Toc38559056)

[1.2.2. Out-of-Scope 5](#_Toc38559057)

[1.3. Test Case Development Approach 6](#_Toc38559058)

[1.4. Test Approach 6](#_Toc38559059)

[1.5. Document Overview 6](#_Toc38559060)

[1.6. Definitions and Acronyms 6](#_Toc38559061)

[2. Requirements 7](#_Toc38559062)

[2.1. Hardware 7](#_Toc38559064)

[2.2. Software 7](#_Toc38559065)

[3. Software Test Approach 7](#_Toc38559066)

[3.1. Unit Testing 7](#_Toc38559068)

[3.2. Smoke Testing 7](#_Toc38559069)

[3.3. System and Integration Testing (SIT) 7](#_Toc38559070)

[3.4. Regression Testing 8](#_Toc38559071)

[3.5. User Acceptance Testing (UAT) 8](#_Toc38559072)

[4. Test Design 8](#_Toc38559073)

[4.1. Test Execution Results 8](#_Toc38559075)

[4.2. Defect Management 8](#_Toc38559076)

[5. Functional Test Cases 9](#_Toc38559077)

[5.1. Test Case 1: Select Procedure 9](#_Toc38559079)

[5.2. Test Case 2: Select Actor 10](#_Toc38559080)

[5.3. Test Case 3: Step Has Warning 11](#_Toc38559081)

[5.4. Test Case 4: Warning Exists for Entire Procedure 11](#_Toc38559082)

[5.5. Test Case 5: Warning Exists for a Set of Steps 12](#_Toc38559083)

[5.6. Test Case 6: Figures Exist for a Step 12](#_Toc38559084)

[5.7. Test Case 7: Long Steps May Not be Broken Over Pages 13](#_Toc38559085)

[5.8. Test Case 8: All Steps are Displayed 13](#_Toc38559086)

[5.9 Test Case 9: Negative Test - Try to go to previous step when on first step 14](#_Toc38559087)

[5.10 Test Case 10: Negative Test - Try to go to next step on Procedure Complete 14](#_Toc38559088)

[5.11 Test Case 11: Negative Test - Try to go to next step without using "Maestro" 15](#_Toc38559089)

[5.12 Test Case 12: Negative Test - Try to choose a procedure that is not displayed 15](#_Toc38559090)

[5.13 Test Case 13: Negative Test - Try to choose a role that is not displayed 16](#_Toc38559091)

[5.14 Test Case 14: Verify each page shows navigation help 16](#_Toc38559092)

[6. Assumptions and Constraints 17](#_Toc38559093)

[6.1. Assumptions 17](#_Toc38559095)

[6.2. Constraints 17](#_Toc38559096)

# Introduction

## Purpose

The purpose of Software Test Plan (STP) is to prescribe the scope, approach, and resources associated with testing activities that will be administrated through the software development life cycle of Heads-Up Display (HUD) system that can be utilized for NASA’s Extra-vehicular Activity/Intra-vehicular Activity (EVA/IVA). This STP document will pinpoints all the times and features that need to be tested to ensure high quality application that meets the business expectations.

## Scope

The STP document will cover the functional and system components of the Heads-Up Display (HUD) system application development. The in-scope and out-of-scope sections below specify the what type of tests will be administrated for the NASA Maestro project.



### In-Scope

The types of tests required for this project will ensure that NASA Maestro project known as Heads-Up Display (HUD) is working properly and meets the business expectation. Testing is very important to ensure high quality software that runs without any issues. This STP document showcases the type of tests required for this project.

* Smoke Testing.
* System and Integration Testing (SIT).
* Regression Testing.
* UAT Testing (Support Only)

The scope of functional testing is derived based on the NASA Maestro HUD system request document. Below functionalities are identified as in scope for this test plan:

* The user launches the application
* The user selects a procedure
* The user selects an operator
* The software then is controlled by voice commands
* “Maestro next step” to go forward
* “Maestro previous step” to go backwards
* Diagrams are displayed on the left-hand side
* Warning are displayed through the process

### Out-of-Scope

* Performance testing is out of this scope for the quality assurance team (QA)

## Test Case Development Approach

Test cases will originate from the project owner’s requirement specified for the Maestro project. All tests cases will be reviewed by the team before execution. The priority of these test cases will be based on their importance and severity.

## Test Approach

The following are the type of tests to be conducted within test execution:

* Unit Testing.
* Smoke Testing.
* System and Integration Testing (SIT).
* Regression Testing.
* UAT Testing (Support Only)

## Document Overview

This document describers the NASA Maestro Heads-Up Display (HUD) system test plan and test cases that ensure high quality software meets the business expectations specified by the product owner. The intended audience of this document includes the project stakeholders to verify the project functional toy and intended, and developer who will create and verify the system functionality.

## Definitions and Acronyms

|  |  |
| --- | --- |
| Abbreviation | Definition |
| E2E | End-to-End Testing |
| SIT | System and Integration Testing |
| UAT | User Acceptance Testing |
| HUD | Heads-Up Display |
| EVA/IVA | Extra-vehicular Activity/Intra-vehicular Activity (EVA/IVA) |
| STP | Software Test Plan |
| SDLC | System development life cycle |

# Requirements



## Hardware

* Desktop/Laptop with Windows 10 operating system, or macOS Mojave

## Software

* Web browser – Chrome and Firefox.
* Visual Studio Code

# Software Test Approach

The software test approach for the STP is developed to document the scope, testing progression, defect management and resource requirements for NASA Maestro HUD development project. This document also describes all the testing involved in this project and deliverable to be produced during system and regression testing of this project.

The test approach is critical to ensure that all major scenarios of the HUD project is covered to ensure high quality product that meets the business expectations.



## Unit Testing

The development team will utilize unit tests to ensure the software is working correctly. These unit tests will be written and maintained by the development team. Each unit test will isolate and test specific functions within the code. Doing so will ensure the respective function works as expected and maintains that expected functionality through future updates.

## Smoke Testing

A quick manual test will be performed when a new functionality is introduced to ensure that major functionality of the software is still working as expected. Blockers or any major bugs should be cleared by this type of testing.

## System and Integration Testing (SIT)

This process pertains to testing the integrated system to ensure that it is still within the scope and the proposed requirements. This document will cover the functional testing of the project.

Approach:

The system and integration tests are both combined for this project. The QA team will ensure that testing planning, test cases execution and reporting activities are covered during this phase.

Entry Criteria:

* All identified smoke tests for this project should pass
* All updates to project documentations are also reflected in this document
* All test cases are linked to business requirement document
* System tests cases are reviewed by the project team
* Project code is unit tested and all failed tests are shared with the QA team

Exit Criteria:

* The completion of system and integration testing – pass rate of 97%.
* All blockers and major defects are fixed and resolved.
* Test report for test cases testing completion.

## Regression Testing

This type of test will be conducted by the quality assurance team to ensure existing functionality was not impacted by code fixes and new enhancements.

Approach:

This ensures that new functionality or a current fix to a functionality has not impacted the rest of the code, and that previous functionality hasn’t changed.

Entry Criteria:

* Regression test cases for affected areas of the project should be ready.
* System and integration testing exit criteria is completed.

Exit Criteria:

* Regression testing has been completed.

## User Acceptance Testing (UAT)

These tests will be performed by the project manager and client team. The UMGC HUD team will provide support for UAT testing and address any quality issues resulting from this testing.

# Test Design



## Test Execution Results

The test execution results showcase failed or passed test cases. All these results are recorded and shared in a test report document through milestone 3 and 4 where development for feature 1 and feature 2 begins.

## Defect Management

A - Defect Identification

All new defects found during test execution shall be logged in as a defect in Jira at https://fezzic.atlassian.net/browse/NMH

B - Defect Severity

The defect severities are as follows:

* ***Severity -1: Blocker*** – This type of defects prevents the application from working. As a result, it must be dealt with immediately.
* ***Severity-2: High*** – This will impair the functionality of the software and must be addressed in higher priority.
* ***Severity-3: Medium*** – This type of defect that may impact certain aspects of the application that considered medium priority.
* ***Severity-4: Low*** – These defects are trivial and can be fixed in a later time.

C - Defect Assignment

Defects are assigned to the project team lead to be reviews and assessed. Then the project team lead will reach out to developers to pinpoint resolution and find causes.

|  |  |
| --- | --- |
| Name | Responsibility |
| Laura Tamshen | QA |
| Ali Alnaqeeb | Backup QA |

# Functional Test Cases

1. Select Procedure
2. Select Actor
3. Step Has a Warning
4. Warning Exists for Entire Procedure
5. Warning Exists for a Set of Steps
6. Figures Exist for a Step
7. Long Steps May Not be Broken Over Pages
8. All Steps are Displayed
9. Negative test - Try to go to previous step when on first step
10. Negative test - Try to go to next step on Procedure Complete
11. Negative test - Try to go to next step without using "Maestro"
12. Negative test - Try to choose a procedure that is not displayed
13. Negative test - Try to choose a role that is not displayed
14. Verify each page shows navigation help

## Test Case 1: Select Procedure

|  |  |
| --- | --- |
| Description: | Upon opening the application, the user is presented with a list of procedures to choose from and may only select one. |
| Requirements: | Based on approved workflow diagram – user must first select a procedure. |
| Prerequisites: | The application is opened and displayed on the HUD. |
| Steps: | 1. User opens the dropdown. 2. User selects a procedure from the list. |
| Expected output: | The system displays a list of procedures upon opening.  The system displays the Select Actor screen after procedure selection. |
| Assumptions: | NA |

## Test Case 2: Select Actor

|  |  |
| --- | --- |
| Description: | After selecting a procedure, the user must select an actor. |
| Requirements: | Create a basic format showing one actor’s steps. |
| Prerequisites: | The application is opened and displayed on the HUD.  The user has selected a procedure. |
| Steps: | 1. User opens the dropdown.  2. User selects an actor from the list. |
| Expected output: | The list contains the correct actors for the selected procedure. That is, only the actors for the given procedure are displayed – not all actors for other/all procedures.  After selection of the actor, the first step for that actor is displayed. |
| Assumptions: | The Select Procedure testcase has completed successfully. |

## Test Case 3: Step Has Warning

|  |  |
| --- | --- |
| Description: | If the currently displayed step has an associated warning, it must also be displayed with the step. |
| Requirements: | Warnings can apply to one step, set of steps, or all steps and must also be displayed. |
| Prerequisites: | The application is opened and displayed on the HUD.  The user has selected a procedure and an actor. |
| Steps: | 1. User selects a step that has an associated warning. |
| Expected output: | The warning text displays with the step text. |
| Assumptions: | The Select Procedure testcase has completed successfully.  The Select Actor testcase has completed successfully. |

## Test Case 4: Warning Exists for Entire Procedure

|  |  |
| --- | --- |
| Description: | If the currently selected procedure has an associated warning, it must also be displayed with every step. |
| Requirements: | Warnings can apply to one step, set of steps, or all steps and must also be displayed. |
| Prerequisites: | The application is opened and displayed on the HUD.  The user has selected a procedure that has warnings associated with the entire procedure and an actor. |
| Steps: | 1. User views each step of the procedure. |
| Expected output: | The warning text displays with each step text. |
| Assumptions: | The Select Procedure testcase has completed successfully.  The Select Actor testcase has completed successfully. |

## Test Case 5: Warning Exists for a Set of Steps

|  |  |
| --- | --- |
| Description: | If the currently selected procedure has a set of steps with associated warnings, the warnings display for each step in the set. |
| Requirements: | Warnings can apply to one step, set of steps, or all steps and must also be displayed. |
| Prerequisites: | The application is opened and displayed on the HUD.  The user has selected a procedure that has warnings associated with some of the steps in the procedure and an actor. |
| Steps: | 1. User views each step of the procedure. |
| Expected output: | The warning text only displays for the appropriate set of steps. Warning does not display for all steps. |
| Assumptions: | The Select Procedure testcase has completed successfully.  The Select Actor testcase has completed successfully. |

## Test Case 6: Figures Exist for a Step

|  |  |
| --- | --- |
| Description: | If the currently selected step has an associated figure, display it with the step. |
| Requirements: | Steps may have reference to figures which would be nice to display as well. |
| Prerequisites: | The application is opened and displayed on the HUD.  The user has selected a procedure that has figure(s) associated with some of the steps in the procedure and an actor. |
| Steps: | 1. User views each step of the procedure. |
| Expected output: | The figure(s) only displays for the appropriate step(s). Figure does not display for all steps.  Note: More than one figure may be associated with a step. |
| Assumptions: | The Select Procedure testcase has completed successfully.  The Select Actor testcase has completed successfully. |

## Test Case 7: Long Steps May Not be Broken Over Pages

|  |  |
| --- | --- |
| Description: | If the currently selected step(s) is long, it must not be broken over pages. |
| Requirements: | Between one and five steps will fit on a page. Most procedures are dozens of steps. Steps must not be broken over two pages. |
| Prerequisites: | The application is opened and displayed on the HUD.  The user has selected a procedure and an actor. |
| Steps: | 1. User views each step of the procedure. |
| Expected output: | Steps are not broken over pages. That is, only the steps that can be fully displayed on a page are displayed per page. |
| Assumptions: | The Select Procedure testcase has completed successfully.  The Select Actor testcase has completed successfully. |

## Test Case 8: All Steps are Displayed

|  |  |
| --- | --- |
| Description: | Verify that all steps per procedure/actor are displayed. |
| Requirements: | Steps must display in HUD. |
| Prerequisites: | The application is opened and displayed on the HUD.  The user has selected a procedure and an actor. |
| Steps: | 1. User views each step of each procedure for each actor. |
| Expected output: | All Steps Are Displayed. That is, no steps are missing.  All steps marked as warning/error display as such. All figures associated with steps are displayed. |
| Assumptions: | NA |

## 5.9 Test Case 9: Negative Test - Try to go to previous step when on first step

|  |  |
| --- | --- |
| **Description:** | Negative test to verify that no action is performed when the user asks to go to the previous step while on the first step of a procedure. |
| **Requirements:** | Steps must display in HUD. |
| **Prerequisites:** | The application is opened and displayed on the HUD.  The user has selected a procedure and an actor. |
| **Steps:** | 1. User views the first step of the procedure.  2. User says “Maestro previous step”. |
| **Expected Result:** | The display remains on the first step of the selected procedure. |
| **Assumptions:** | **NA** |

## 5.10 Test Case 10: Negative Test - Try to go to next step on Procedure Complete

|  |  |
| --- | --- |
| **Description:** | Negative test to verify that no action is performed when the user asks to go to the next step while on the Procedure Complete screen of a procedure. |
| **Requirements:** | Steps must display in HUD. |
| **Prerequisites:** | The application is opened and displayed on the HUD.  The user has selected a procedure and an actor. |
| **Steps:** | 1. User navigates to the final step and waits for the Procedure Complete screen to display.  2. User says “Maestro next step”. |
| **Expected Result:** | The application ignores the command and goes to the Select Procedure screen. |
| **Assumptions:** | **NA** |

## 5.11 Test Case 11: Negative Test - Try to go to next step without using "Maestro"

|  |  |
| --- | --- |
| **Description:** | Negative test to verify that no action is performed when the user does not use the trigger word “Maestro”. |
| **Requirements:** | Steps must display in HUD. |
| **Prerequisites:** | The application is opened and displayed on the HUD.  The user has selected a procedure and an actor. |
| **Steps:** | 1. User navigates to the first step of the procedure  2. User says “next step”. |
| **Expected Result:** | The application ignores the command and remains on the current screen. |
| **Assumptions:** | **NA** |

## 5.12 Test Case 12: Negative Test - Try to choose a procedure that is not displayed

|  |  |
| --- | --- |
| **Description:** | Negative test to verify that no action is performed when the user attempts to select a non-existent procedure. |
| **Requirements:** | Steps must display in HUD. |
| **Prerequisites:** | The application is opened and the Select Procedure screen displays. |
| **Steps:** | 1. User uses voice command to attempt to select a procedure that does not exist. (e.g. “Maestro select procedure xxx”.) |
| **Expected Result:** | The application remains on the Select Procedure screen. |
| **Assumptions:** | **NA** |

## 5.13 Test Case 13: Negative Test - Try to choose a role that is not displayed

|  |  |
| --- | --- |
| **Description:** | Negative test to verify that no action is performed when the user attempts to select a non-existent role. |
| **Requirements:** | Steps must display in HUD. |
| **Prerequisites:** | The application is opened and the user has selected a procedure. |
| **Steps:** | 1. User uses voice command to attempt to select a role that does not exist. (e.g. “Maestro select role xxx”.) |
| **Expected Result:** | The application remains on the Select Role screen. |
| **Assumptions:** | **NA** |

## 5.14 Test Case 14: Verify each page shows navigation help

|  |  |
| --- | --- |
| **Description:** | Test to verify that each page of displays navigation help. Note that this was a late requirement and may not be implemented by project end. |
| **Requirements:** | Each screen shows navigation help. |
| **Prerequisites:** | The application is opened and the user has selected a procedure and role. |
| **Steps:** | 1. User uses voice commands to navigate through a procedure. |
| **Expected Result:** | Each page shows navigation help. |
| **Assumptions:** | **NA** |

# 6. Assumptions and Constraints



## Assumptions

This project scope is limited to the activities required to develop Heads-On Display (HUD) interface for NASA Maestro project. Based on the existing Maestro application, we assume the following:

* The GitLab repository (Repo) is set up to detect any code changes and trigger a build
* When committing and pushing an update of the source file into the Maestro project Repo, it will generate a build as an output
* All NASA team members will be involved through the SDLC of the project
* Dependencies identification to ensure facilitation of Maestro project development pull and push files from the Repo, as well as creating, editing, and updating YAML files.

## Constraints

The following are some of the limitations which may impede project success:

* Building on an existing Maestro application would require some knowledge regarding the pre-existing technologies used to speed up the development process
* The application would require a server to run and that will be of a cost to the team
* Pre-existing or unknown bugs which might derail the development process at the beginning
* Certain Maestro project features might not be fully developed during the previous phase of this project.
* Daily communication, and real time cooperation within a time that are located within different time zones can be very challenging.