**Software Project Management Plan**

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

Version 4

April 26, 2020

Prepared by Ali Alnaqeeb

Team Members

Ali Alnaqeeb

Laura Tamshen

Ryan Cohan

Matthew Elliott

Jonathan Johnson

**Index of Revisions**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Name | Date | Description |
| 1.0 | Ali Alnaqeeb | 02/15/2020 | Initial draft and structure set up |
| 1.1 | Ali Alnaqeeb | 02/28/2020 | Update Stakeholders section |
| 1.2 | Ryan Cohan | 02/28/2020 | Review and reformat |
| 1.3 | Ali Alnaqeeb | 02/28/2020 | Update workflow diagram |
| 1.4 | Ali Alnaqqeb | 03/09/2020 | Added RACI matrix, added project communication matrix, add more information to agile milestones, update sections descriptions, updated platform to Jira, testing and agile methodology. |
| 2 | Ali Alnaqeeb | 3/21/2020 | Update the entire document to correspond with SRS, SDD, and test reports, updated images, burndown chart and formatting. |
| 2.1 | Ali Alnaqeeb | 3/25/2020 | Updated Risks, RACI Matrix, Impact Issues, Team Formation, Sprint burndown charts and update Scope steps. |
| 3 | Ali Alnaqeeb | 04/05/2020 | Added project status section and updated risk section |
| 3.1 | Ali Alnaqeeb | 04/08/2020 | Updated Milestone 4 WPS, RACI Matrix, and milestone 4 submission list |
| 4 | Ali Alnaqeeb | 04/24/2020 | Finalize document with all updates |

Contents

[1. Overview 7](#_Toc36983149)

[1.1. Purpose 7](#_Toc36983150)

[1.2. Project Background 7](#_Toc36983151)

[1.3. Scope 7](#_Toc36983152)

[1.4. Goals and Objectives 8](#_Toc36983153)

[1.5. Project Deliverables 8](#_Toc36983154)

[1.6. Assumptions and Constraints 11](#_Toc36983155)

[1.6.1. Assumptions 11](#_Toc36983156)

[1.6.2. Constraints 11](#_Toc36983157)

[1.7. Schedule 11](#_Toc36983158)

[1.7.1. Milestone 1 11](#_Toc36983159)

[1.7.2. Milestone 2 12](#_Toc36983160)

[1.7.3. Milestone 3 12](#_Toc36983161)

[1.7.4. Milestone 4 13](#_Toc36983162)

[1.8. Milestone Burndown Charts 13](#_Toc36983163)

[1.8.1. Milestone 1 Burndown Chart 13](#_Toc36983164)

[1.8.2. Milestone 2 Burndown Chart 14](#_Toc36983165)

[1.8.3. Milestone 3 Burndown Chart 15](#_Toc36983166)

[1.8.4. Milestone 4 Burndown Chart 16](#_Toc36983167)

[1.9. Success Criteria 16](#_Toc36983168)

[1.10. Definitions 16](#_Toc36983169)

[2. Stakeholder Management 17](#_Toc36983170)

[2.1. External Interfaces 17](#_Toc36983171)

[2.1.1. NASA 17](#_Toc36983172)

[2.1.2. Internal Structure 17](#_Toc36983173)

[2.2. Project Roles and Responsibilities 17](#_Toc36983174)

[2.2.1. Project Sponsor 17](#_Toc36983175)

[2.2.2. Instructor 18](#_Toc36983176)

[2.2.3. Project Team 18](#_Toc36983177)

[2.2.4 Team Adjustments 18](#_Toc36983178)

[2.2.5 Impact Issues 18](#_Toc36983179)

[2.2.4. RACI Matrix 19](#_Toc36983180)

[2.3. Project Communications 20](#_Toc36983181)

[2.3.1. Project Communication Matrix 20](#_Toc36983182)

[2.3.2. Electronic Media 21](#_Toc36983183)

[2.3.3. Meetings 21](#_Toc36983184)

[2.3.4. Project Closeout 21](#_Toc36983185)

[2.4. Project Initiation 22](#_Toc36983186)

[2.5. Tools and Technologies 22](#_Toc36983187)

[3. Work Plan 22](#_Toc36983188)

[3.1. Software Development Methodology – Agile Scrum 22](#_Toc36983189)

[3.2. Milestones 23](#_Toc36983190)

[3.3. Weekly Report 23](#_Toc36983191)

[3.4. Milestone Report 23](#_Toc36983192)

[3.5. Estimation Refinements 23](#_Toc36983193)

[3.6. Workflow 23](#_Toc36983194)

[4. Control Plan 24](#_Toc36983195)

[4.1. Monitoring and Control 24](#_Toc36983196)

[4.1.1. Unit Testing 24](#_Toc36983197)

[4.1.2. Continuous Integration 24](#_Toc36983198)

[4.2. Configuration Management Plan 24](#_Toc36983199)

[5. Supporting Process Plan 25](#_Toc36983200)

[5.1. Risk Management Plan 25](#_Toc36983201)

[5.2. Risk Identification 26](#_Toc36983202)

[5.2.1. Risk Analysis 26](#_Toc36983203)

[5.2.2. Risk Response Planning 27](#_Toc36983204)

[5.2.3. Risk Monitoring and Tracking/Risk Log 27](#_Toc36983205)

[5.3. Risk Reporting 27](#_Toc36983206)

[5.4. Test Plan 27](#_Toc36983207)

[5.5. Scope 27](#_Toc36983208)

[5.5.1. In-Scope 27](#_Toc36983209)

[5.5.2. Out-of-Scope 27](#_Toc36983210)

[5.6. Test Case Development Approach 28](#_Toc36983211)

[5.6.1. Test Approach 28](#_Toc36983212)

[5.6.2. Smoke Testing 28](#_Toc36983213)

[5.6.3. System and Integration Testing (SIT) 28](#_Toc36983214)

[5.6.4. Regression Testing 28](#_Toc36983215)

[5.7. Defect Management 28](#_Toc36983216)

[5.7.1. Defect Severity 29](#_Toc36983217)

[5.7.2. Defect Assignment 29](#_Toc36983218)

[5.8. Project Status 29](#_Toc36983219)

[6. References 29](#_Toc36983220)

# Overview

## Purpose

Our customers at National Aeronautics and Space Administration (NASA) are requesting the development of a Heads-Up Display (HUD) system that can be utilized for NASA’s Extra-Vehicular Activity/Intra-Vehicular Activity (EVA/IVA). The display has a 4-inch screen size with 480x800 pixel resolution. The astronaut procedure steps will be shown on this display. This HUD will also be programmed so that it can interact with voice commands. The voice commands that will control this software are “Maestro next step” and “Maestro previous step” as well as “Maestro show image [image number]” to enlarge. The user of the software will be able to select which actor he/she will be acting as, which - since procedures are a compilation of many different steps for many different actors – dictates which steps will be displayed on the HUD.

## Project Background

Currently, the employees at NASA create spacewalk procedures known as EVAs. Typically, these procedures include two astronauts that are located outside of two stations known as EV1 and EV2. In addition, there is a robotics operator inside the spacecraft and Mission Control on the ground assisting in the walk.

There are situations in which EV1 is focused on a task where all the steps can be done independent of EV2’s actions. There are other times when their steps are heavily intertwined. The same goes for robotics operators and Mission Control steps. The customers at NASA have a need to programmatically parse procedures, map each step to its respective actor, and display this on a hands-free screen.

Maestro currently renders procedure data in many file types (HTML, DOCX, specific XML types) as well as different formats (EVA format, IVA format, etc.). The project is essential for fulfillment of the UMGC Spring 2020 SWEN-670 Software Engineering Project.

## Scope

After meeting with the project manager and client team, the following requirements were elicited:

1. Launch the Maestro application
2. Load a list of procedures for the user to select from using voice command ***Maestro select procedure (number)***
3. After selecting a procedure, user will be presented with a list of associated roles to select from using voice command ***Maestro select role (number)***
4. The system displays the first step of the first task, as well as any images, associated with the selected procedure
5. User can navigate through the procedure with the following voice commands:
   1. ***Maestro Select Procedure (procedure number) ­***– selects the supplied procedure number
   2. ***Maestro Select Role (role number)*** – selects the supplied role number and displays the first associated task step
   3. ***Maestro Next Step*** – shows the next step in the task
   4. ***Maestro Previous Step*** – shows the previous step in the task
   5. ***Maestro Go Home*** – navigates back to the landing page where the user can select a new procedure
   6. ***Maestro Show Image (image number)*** – enlarges the selected image number
   7. ***Maestro Back to Step*** – exits the current, enlarged image and returns to the step
   8. ***Maestro Next Image Page*** – shows the next set of images; does not change task step
   9. ***Maestro Previous Image Page*** – shows the previous set of images; does not change the task step
   10. ***Maestro Toggle Checkbox (checkbox number***) – toggles the selected checkbox to either selected or unselected
   11. ***Maestro Help*** – brings up the help screen listing all voice commands
6. When the user advances past the last page of steps for the final task associated with a procedure, the UI should display a screen with the message “Procedure Complete”. After 5 seconds, the UI should be redirected to the procedure selection screen
7. All this information is displayed in a small 4” display located in the astronaut helmet

## Goals and Objectives

The goal and objective of Nasa Maestro is to replace the manual procedure of creating procedures for Extra-vehicular Activity (EVA) with a tool able “to write EVA procedures as simple YAML files and run build to generate Word document(s) in the standard procedure format”. As a result, the goal of the NASA Maestro HUD project is to develop an interface that will parse and render those procedures in a 4” display. The display will fit inside the astronaut’s helmet and will display the steps taken outside of the EV1 and EV2 stations. The object is to enhance the user experience and facilitate initial bootstrapping, structural changes, editing and updates of procedures and tasks source files needed to automate the generation of EVA tasks artifacts.

## Project Deliverables

A summary of project deliverables is shown in Table 1.

|  |  |  |
| --- | --- | --- |
| Deliverable | Description | Remark |
| Milestone 1 | | |
| Team NASA 2 Roster | Names, Role assignments and contact info for the team |  |
| System Request | Provides the basic information and analysis regarding the proposed system and project value |  |
| Milestone and Sprint Planning | A breakdown of all milestones and sprints due dates |  |
| Software Project Management Plan (PMP) | Document to provide guidance and control of the Heads-On Display application software project. | Initial Draft |
| Software Requirement Specifications (SRS) | This Software Requirements Specification (SRS) describes requirements for the NASA Maestro HUD project | Initial Draft |
| Milestone 1 Presentation | Presentation highlighting the work accomplished during each milestone, problems encountered and actions for improvement |  |
| Milestone 2 | | |
| Test Planning Document | Describes the test approach, type of tests for feature 1 project | Initial Draft |
| Test Report | A summary of executed tests and results | Initial Draft |
| Project Communication Matrix | Showcases the project team communication platforms and meetings frequency. | Initial Draft |
| Risk Matrix | describes the implication of how new codes might impact previous functionality. | Initial Draft |
| SDD, SRS | Describes system architecture and requirements |  |
| Milestone 3 | | |
| Test Plan, Test Report Documents | Describes the test approach, type of tests for and test plan for NASA Maestro project | Version 3 |
| SRS | Updated version of the SRS regarding requirements | Version 3 |
| SDD | Update version of SDS regarding system architecture and design | Version 3 |
| Project Plan | Updated version of project plan | Version 3 |
| User Guide | Describes how to use the system | Version 3 |
| Runbook | Describes how the system is used by the system admin | Version 3 |
| Milestone 4 | | |
| Test Plan, Test Report | Describes the test approach, type of tests for and test plan for NASA Maestro project | Ver 4 |
| SRS | Describes NASA Maestro HUD requirements | Ver 4 |
| SDD | Describes NASA Maestro HUD system architecture and design | Ver 4 |
| Project Plan | Describes the process to timings and project direction of developing Maestro HUD system | Ver 4 |
| User Guide | Describes how to use the system | Ver 4 |
| Project Closure | Describes the project status and information | Ver 4 |
| Milestone 4 Presentation | A Final presentation and wrap up for the project | Final |
| Runbook | Describes system usage by admin | Ver 4 |

Table 1. Project Deliverables

## 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 GitHub 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:

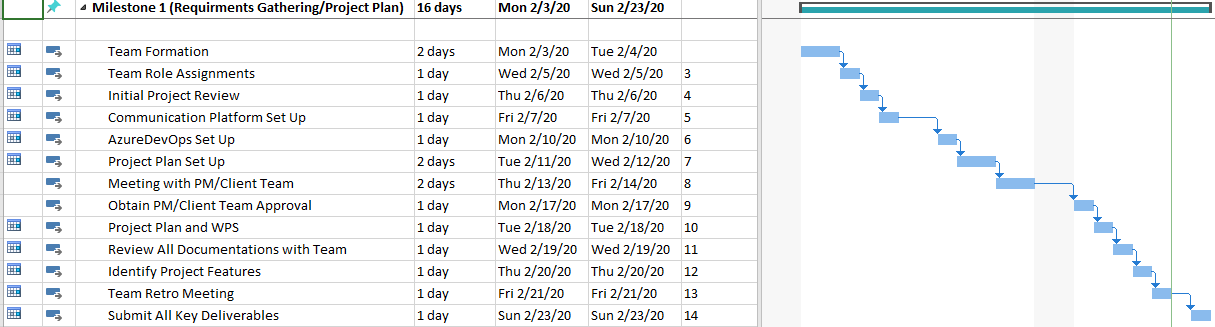
* Building on an existing Maestro application would require some knowledge regarding the pre-existing technologies used to speed up the development process
* The application would require a server to run and that will be of a cost to the team
* Pre-existing or unknown bugs which might derail the development 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.

## Schedule

The NASA Maestro Project schedule is broken into four major milestones. A summary of the schedule is shown below, along with Gantt Charts broken out for each Milestone indicating key activities and tasks.

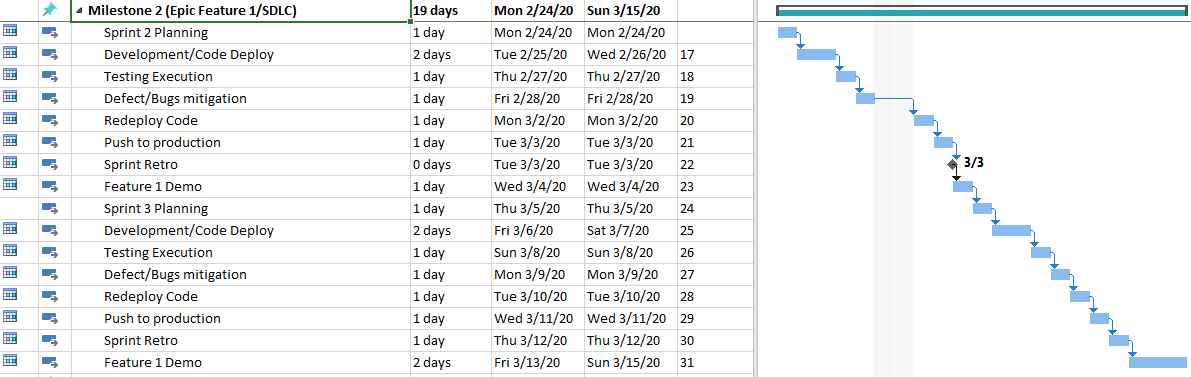
### Milestone 1

The Gantt chart below describe Milestone 1 activities. These activities involved team set up, planning, requirement analysis, project plan initiation, and system request approval.



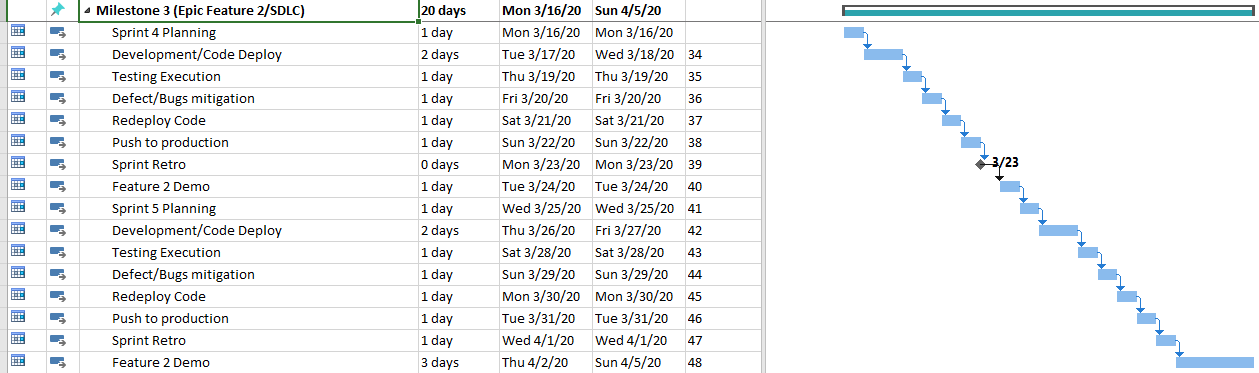
### Milestone 2

The Gantt chart below describe Milestone 2 activities. These activities application development, test planning, testing reports, and documentations updates.



### Milestone 3

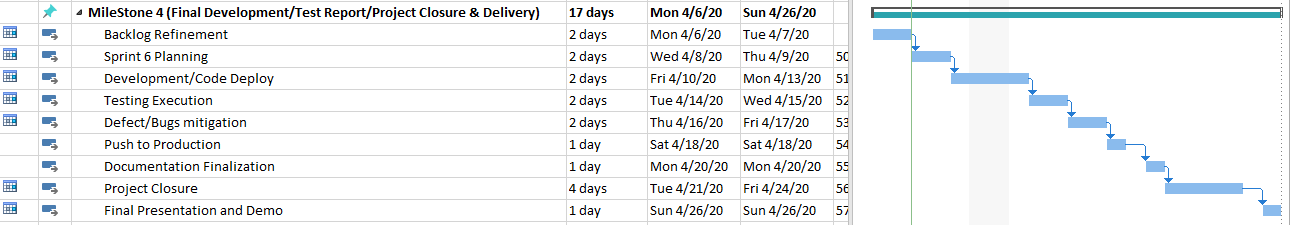
The Gantt chart below describe Milestone 3 activities. These activities application development, test planning, testing reports, and documentations updates.



### Milestone 4

The Gantt chart below describe Milestone 4 activities. These activities application development, test report, project closure report, and milestone 4 presentation.

Since the risks associated with the NASA Maestro HUD project has changed, affecting the progress of the project, the team will utilize the final milestone to conduct development as well. As a result, this milestone will be a wrap up for the entire project. Activities in this project will be development, testing, documentation, project delivery and final application demo. Please refer to section 2.2.5 Impact issues. The following WPS is updated accordingly:



## Milestone Burndown Charts

### Milestone 1 Burndown Chart

Milestone 1 included only Sprint 1. This sprint included tasks related to documentation such as the Project Plan, SRS, etc. Below is the burndown chart that highlights the activities that were completed during sprint 1. Unfortunately, there was an issue with Jira where tasks didn’t have Story Points, as a result, the sprint 1 burndown chart didn’t get updated. However, using an excel sheet, a manual presentation of sprint 1 can be shown in figure 2 below.

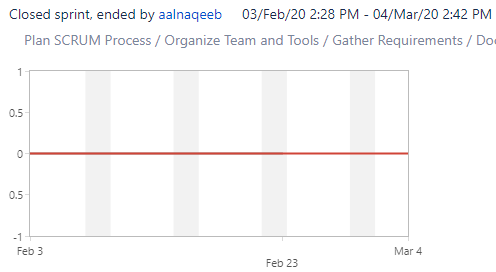


Figure 1 – Sprint 1

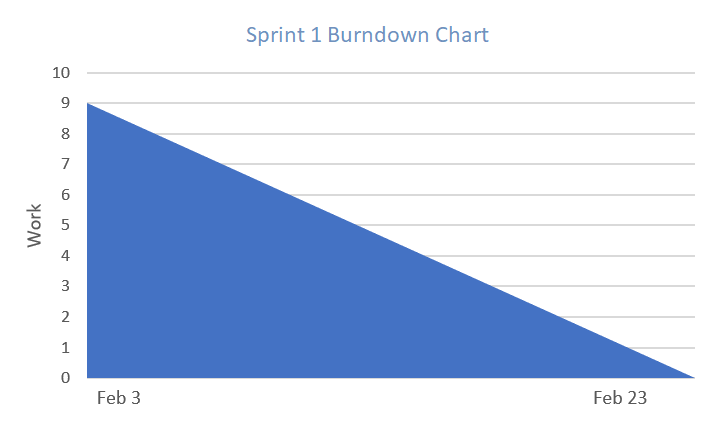
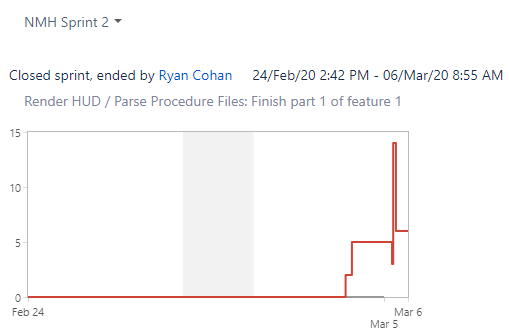


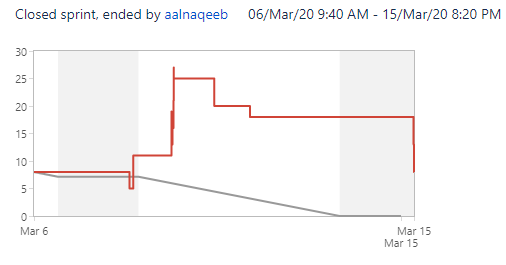
Figure 2 – Excel Sprint 1

### Milestone 2 Burndown Chart

Milestone 2 included two sprints. Sprint 2 and Sprint 3. These sprints included coding tasks as well as documentation tasks such as Test Report and updated Project Plan.



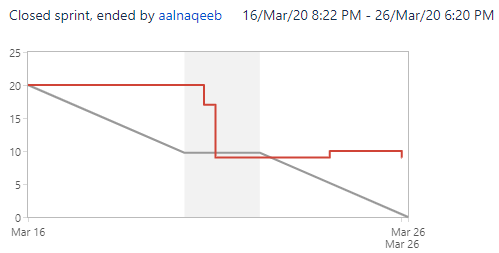
Sprint 2



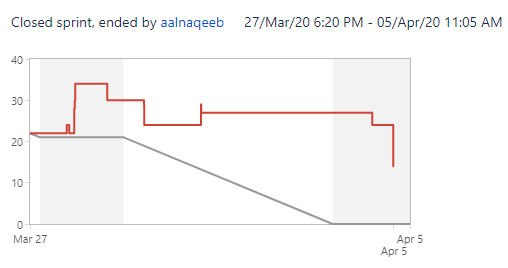
Sprint 3

### Milestone 3 Burndown Chart

Milestone 3 included sprint 3 and sprint 4. In these sprints, the team continue to build the application by parsing the information from the YML file – backend and connecting it with the HUD. These sprints also included documentation, in addition to testing, fixing defects, and new team formation.



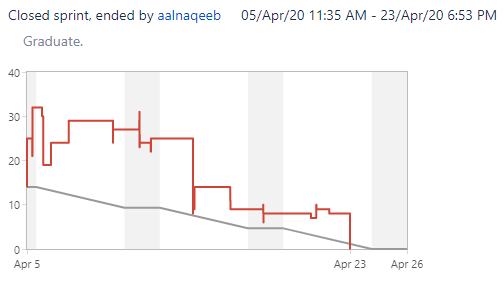
Sprint 4



Sprint 5

### Milestone 4 Burndown Chart

Milestone 4 included sprint 6. In this sprint, the team focused on wiring the backend to frontend. This sprint also included documentation, in addition to testing, fixing defects, and deployment.



Sprint 6

## Success Criteria

Feature 1 and 2 are completed per requirements specified by the project manager.

## Definitions

HUD – Heads-On Display

IVA - Intra-vehicular Activity

EVA – Extra-vehicular Activity

NASA - National Aeronautics and Space Administration

PMP – Project Management Plan

SRS – Software Requirements Specification

SWEN 670 – Software Engineering course number 670

UMGC – University of Maryland Global Campus

WIP – Work in Progress

Repo – Repository

# Stakeholder Management

The stakeholders represent a critical part of the software development process for the NASA project. Their input determines how the software will be shaped and if there are any issues that need to be solved. They exist both within and outside the organization. The stakeholders for NASA Maestro project consist of the following:

* UMGC Team (See 2.2.3 Project Team section for team list)
* Two DevOps engineers who will provide our Developers assistance with development tools and infrastructure
* Two product owners - James Montalvo and Kristopher Field
* Project manager Roy Gordon
* The astronauts and operators who will be using the NASA Maestro HUD software
* Dr. Assaduallah who is the professor for our class

## External Interfaces

### NASA

To ensure the team is on the same page, various meetings will be conducted using video chat platforms to review documentation and discuss requirements. This will allow for rapid requirements elicitation and refinement, as well as feedback on developed items.

### Internal Structure

The NASA team consists of 5 members of the 670 SWEN 2020 Sprint semester class. To ensure successful execution of this project, the team must maintain frequent communications and up-to-date progress using the chosen task management software (Jira). They will share equal responsibility of the project outcome and will equally contribute to its success.

## Project Roles and Responsibilities

### Project Sponsor

James Montalvo, Kristopher Field (NASA)

### Instructor

Dr. Mohammad Assadullah

### Project Team

The NASA Maestro Team of Project 2 consists of the UMGC Software Engineering graduate students listed below. Members of the project team function as peers and share responsibility for all aspects of the project including management, documentation, implementation, and testing.

### Team Adjustments

Team 2 NASA Maestro team formation has been adjusted recently. Due to the crisis of the COVID-19, the professor has requested from this project students to volunteer to design a ventilator machine to help with the patients who are having difficulties breathing. As a result, one of our team members – Ryan Cohan who is our lead backend developer has volunteered to help with the new project since 3/23/2020. Our team now consist of Ali, Laura, Matt, and Jonathan and Ryan on a part-time basis. Please see 2.2.5 for the impact this change has caused to the team.

### Impact Issues

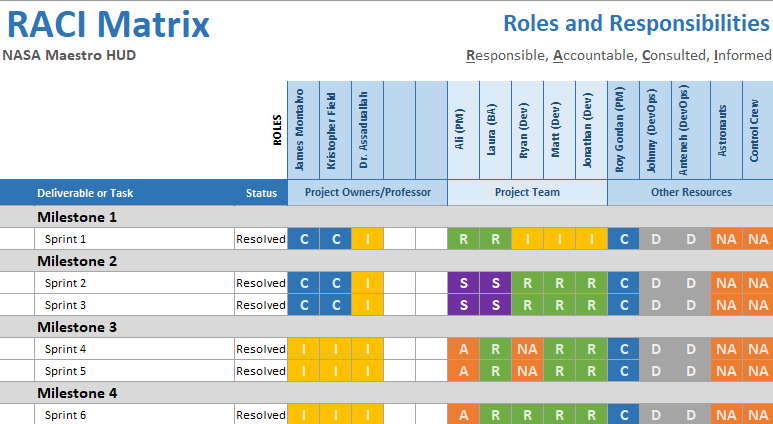
This change has added more risks to the NASA Maestro project. Please see 5.1 Risk Management Plan for an updated list of risks this impact has caused. Issues and expectations were shared with the product owners James and Kris. They are aware of the new change and their expectations of the project completion and number of artifacts submitted are considered accordingly. Also, Mr. Roy, who is the project manager has been consulted. Dr. Assaduallah understand this impact as well and will also consider the risks this change has caused to the project deadline and artifacts submission.

As of 8:20 PM EDT April 7th, 2020, Kris and James (NASA Product Sponsors) have approved that we’ll not deliver a developer’s guide due to the lack of resources. This change has also been communicated to Mr. Roy (Project Manager), and Dr. Assadullah (Professor).

|  |  |
| --- | --- |
| Team Members | Role(s) |
| Ali Alnaqeeb | Team Lead/Project Manager |
| Laura Tamshen | Business Analyst/Tester |
| *Ryan Cohan* | *Developer/Systems Architect (Part time)* |
| Matthew Elliott | Developer |
| Jonathan Johnson | Developer |
| DevOps Support | Role(s) |
| Johnny Lockhart | DevOps |
| Haile, Anteneh | DevOps |

### RACI Matrix

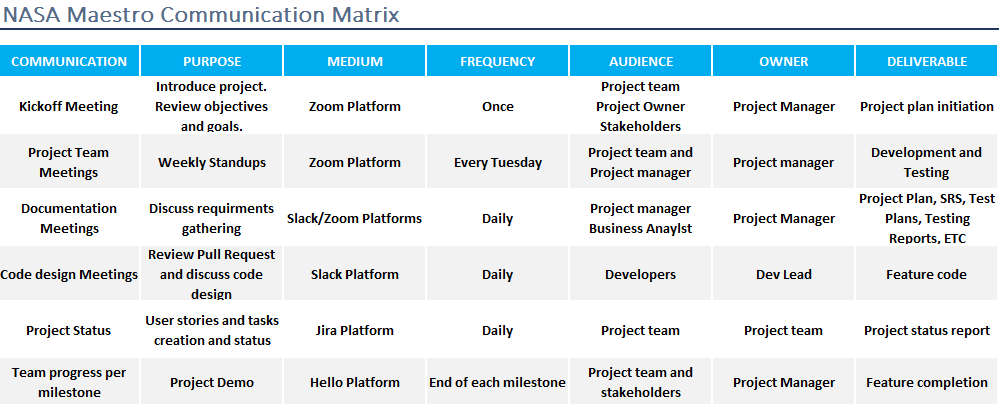
The RACI is acronym that stands for Responsible, Accountable, Consulted, and Informed. The RACI matrix will be used to provide high-level progress report regarding each member of the stakeholders for NASA Team 2. Alongside the progress report this matrix provides, it also shows how each member of the team, including stakeholders, are addressed throughout the SDLC. For Sprint 1 which was mainly documentation and team set up, the RACI shows C for the product owners, as consulted, and the professor was informed with the status. The team project manager and business analyst were marked as R which were responsible for leading the sprint. The team developers were informed of the team set up and the direction the project is heading so they were marked as I for informed. The NASA project manager was consulted through this process and marked as C. Finally, the DevOps were marked as D for driver which they assisted in the project pipeline set up and streamlined role permissions for team developers.



## Project Communications

### Project Communication Matrix

This is the project communication matrix that will be utilized in this project. Since the stakeholders are dispersed around the country in different time zones, the team project manager decided to create a project communication matrix. This matrix showcases the communication settings the team is following to keep the project going in the right direction. Communication is the key to having a successful project. Daily meetings such as documentation set up, code design, and project status are held among the team members through Slack, Zoom and Jira Platforms. Other meetings regarding kickoff meeting and project demos are done using Hello Platforms since they usually go over the Zoom meeting time limitation. Below is the Matrix for our team:



### Electronic Media

The team has decided to use Slack as its communication method and Zoom platform to set up conference calls with the project manager, client team, and class professor. Sometimes, the team needs more time to clarify few things regarding code, documentation, or requirements with the professor, or product owners, as a result, Hello platform is used for these types of meetings. This platform provides unlimited phone calls, and most importantly, screen sharing.

### Meetings

The NASA project team will meet once a week each Saturday from 8 to 9 PM EST. This meeting is essential as a daily standup format meeting to share what the team did last week, what are they currently working on, and if there are any blockers. The team is currently using Zoom for these types of meetings. As needed, the project manager or any of the team members will request additional meetings as deemed necessary. In addition to the weekly status meeting, there are backlog refinement meetings that happen every sprint to go over and evaluate the user stories. The project team will meet at the end of each sprint to review progress, priorities, and tasks. We also discuss certain users’ stories that need to roll over to the next sprint. The project sponsor and instructor are invited to attend these meetings, but their attendance is not required. The team will also have a weekly stand up with project manager so that he can be briefed of the team’s progress. This ensures the team direction is accurate and aligns with stakeholder needs.

### Project Closeout

The key deliverables for this project will be submitted by the team leader to the instructor through the UMGC LEO assignment section at each milestone. Other artifacts of the project, such as source code, will be maintained on GitHub created and maintained by the instructor in coordination with the DevOps team.

## Project Initiation

The project need was emphasized by the sponsor who was seeking ways of developing a web interface that would enable authors of NASA procedures to display astronaut steps in a 4” display HUD.

## Tools and Technologies

* Slack for collaboration and team communication
* Zoom
* Hello Conference Call
* Google Docs for documentation
* M.S Project
* M.S Word
* M.S PowerPoint
* GitHub for version control and source code repository
* Visual Studio
* Jira DevOps for tracking project workflow and product backlog

# Work Plan

## Software Development Methodology – Agile Scrum

Agile methodology is an iterative software development approach that helps the team to deliver the product to their customers. This process, as opposed to waterfall, delivers the work in small but consumable increments. Scrum on the other hand is a subset of Agile methodology. Scrum is a lightweight process framework for agile development, and the most widely-used one. Please refer to scrum.org for implementation details.

The methodology used in this project will be Scrum. A backlog is created with a list of tasks that fall under feature 1 and feature 2 to be prioritized and planned throughout the software development life cycle (SDLC). These tasks are all maintained on Jira. They are created by the business analyst and are assigned to developers accordingly.

The first milestone includes project management and SRS documentation. The second and third milestone will focus on the software development life cycle of the Maestro HUD project. Lastly, fourth milestone will focus on finishing up missing items and enhancements, documents updates, and presentation.

## Milestones

The milestones in the Maestro projects are iterative and incremental throughout the SDLC. The development team will demo the work accomplished at the end of each milestone and elicit stakeholder feedback. This feedback will be captured and incorporated into future sprints for development.

## Weekly Report

The project manager will present the product manager and owners with a weekly report of progress, questions, and impediments.

## Milestone Report

At the end of each milestone, the project team will meet with NASA product manager and client team to review the developed features. These features can be updated in the upcoming milestones.

## Estimation Refinements

Based on the product manager and client team feedback, features can be added, removed or enhanced. Issues that occurred in previous sprints are taken into consideration when refining the new estimates

## Workflow

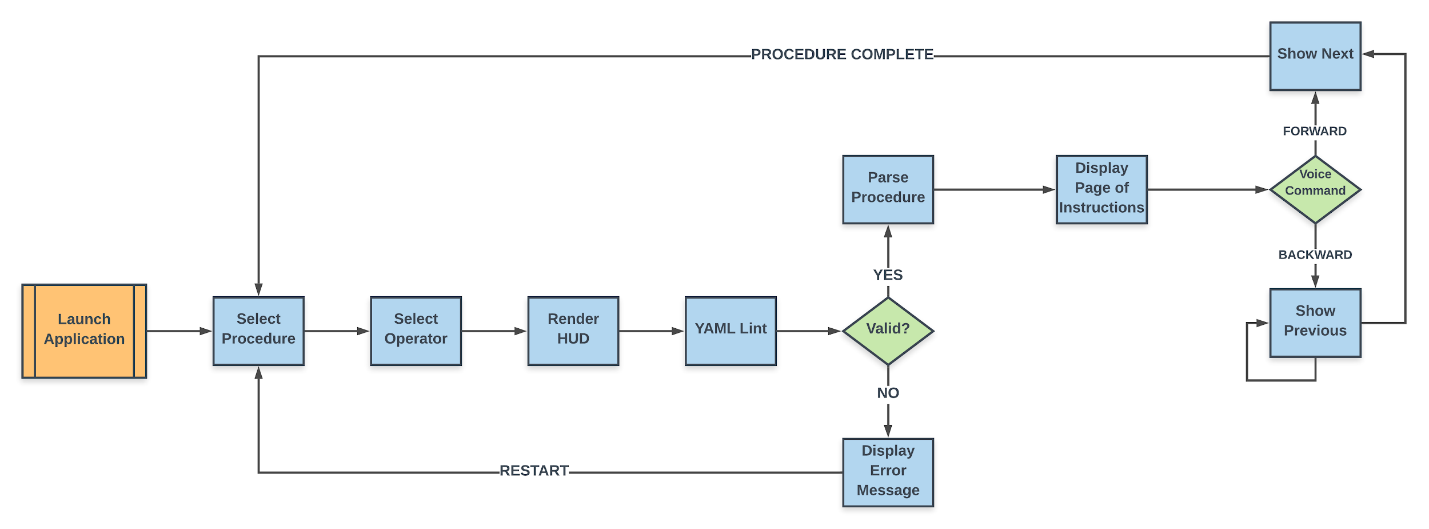


Figure 1 - Project Workflow Version 2

The following describes the figure above and is the project work items process flow from requirement definition to testing:

* 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

# Control Plan

## Monitoring and Control

The project requirements official document will be all kept in the software requirement specifications document (SRS). The document is updated as the team progress in the development process. The team will also conduct unit tests, continuous integration, and configuration management to ensure the continuous stability and high quality of the project. Because the application is controlled by voice commands, test automation for the front end is not feasible. Therefore, manual testing will be performed on the front-end. In addition to that, chrome DevTools will be used to ensure that the application is responsive in a 4-inch display. Automated testing will be performed in the form of unit and continuous integrating testing as described including in section 4.1.1, and 4.1.2.

### Unit Testing

The development team will unitize unit tests to ensure the software is working correctly. Also, these units’ tests can detect any issues when integrated into the main product.

### Continuous Integration

The integration in Maestro project will be continuous so that all changes don’t remove previous functionality from the product. The project’s GitHub will be monitored for changes and all the updates to the project will be built and tested to ensure continuing functionality. This automatic process will ensure broken or bugged builds are detected early. Tests will be run after each commit to ensure previous functionality hasn’t changed.

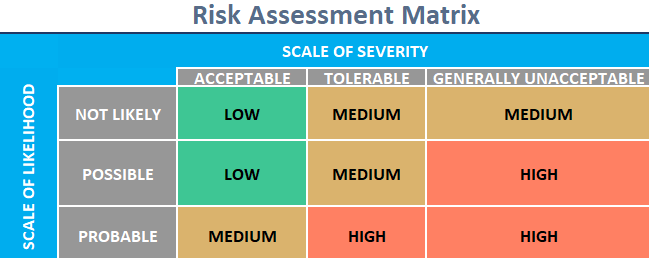
## Configuration Management Plan

The NASA Maestro HUD project will use standard configuration management processes. All changes to the software will be approved by Change Control Board (Project team, project managers, and stakeholders) for final approval. As soon as these changes are approved, then they can be approved to the project. Change Request Management process depends on the type of change requested. If the team requests a change in the feature, then the team reaches out to the product owners to get that request approved, otherwise, the team will follow the requirement stated in the project description.

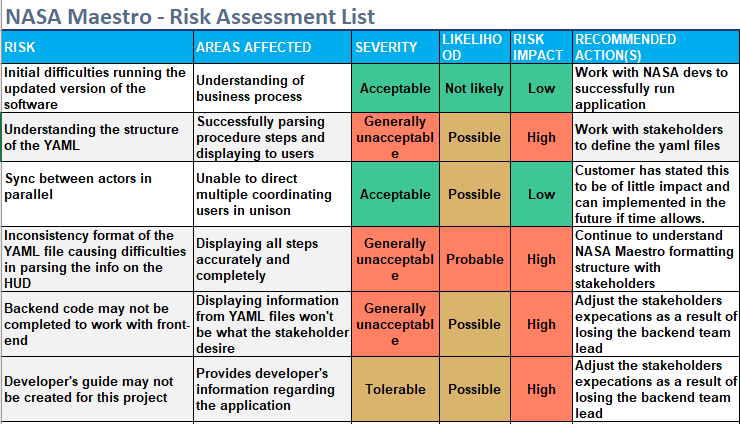
# Supporting Process Plan

## Risk Management Plan

The NASA team are responsible for identifying and managing any risks that are associated with the development of Maestro HUD project. The project manager and client team are also encouraged to shed their light on the development process and highlight any risks to the team project. This is very important to avoid any issues and possibly prevent any future bugs. The risk management plan also contains the Risk Matrix below:



And the Risk Register list below. This list shows a list of risk and the areas affected by these risks. This table also has other columns such as the severity of the issue, the possibly of these issues occurring, risk impact and the recommended actions to mitigate these issues.



## Risk Identification

Identifying risks that are associated with Maestro project may positively or negatively impact the project, documenting and describing their characteristics. This process just like agile is an iterative process which occurs first during project planning phase, and throughout the SDLC. The team is responsible for capturing any anticipated risks and escalating them to the project manager and client team. As the project evolves, and more features are added, the entire team including the stakeholders will also have the opportunity to identify possible risks. The product owners’ feedback will be very important with risk identification as they know their software more than anyone else on the team.

### Risk Analysis

If risks are found, they will be kept and maintained in a risk log. The project team will perform the following and analyze these risks:

* Perform risk and probability assessment
* Risk quality assessment.
* Risk categorization.
* Perform qualitative and quantitative risk assessment.
* Plan for contingencies in the event of risk occurrence.

### Risk Response Planning

The team is aware that this project might bring an unexpected risk. The plan is to conduct meetings to take consideration for risk transfer, risk mitigation and risks acceptance.

### Risk Monitoring and Tracking/Risk Log

Any new risks captured by the project team are reported and escalated to the project manager and client team. These risks are kept and maintained in a risk log.

## Risk Reporting

NASA Maestro project will have a risk management report that will be reviewed with the project team weekly and kept up to date.

## Test Plan

Test plans will be delivered within milestones 2, 3 and 4. The test plan will be based upon requirements given from the product owners. Test planning is essential to ensure test coverage for all requirements has been established. They will be updated each milestone based upon completed stories. After discussion with developers, the QA team is more familiar with what need to be tested and how.

## Scope

This test plan covers all the functional aspects of the maestro project related to the HUD functionality. This section goes into the in-scope and out-of-scope functionality of the planned for this effort.

### In-Scope

Testing in the Maestro project will cover the development phase of the SDLC. Testing is done to ensure the software has sustained its quality and is to be usable by the client team. The type of tests will be smoke tests, functional tests, and regression tests.

### Out-of-Scope

The tests that are out of scope for this project is performance and UAT testing. These tests will be performed by the project manager and client team.

## Test Case Development Approach

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

### Test Approach

Smoke testing, system and integration testing, and regression testing are the type of tests to be conducted within test execution.

### Smoke Testing

A quick test that will be performed when a new functionality is introduced to ensure that major functionality of the software is still working as expected.

### System and Integration Testing (SIT)

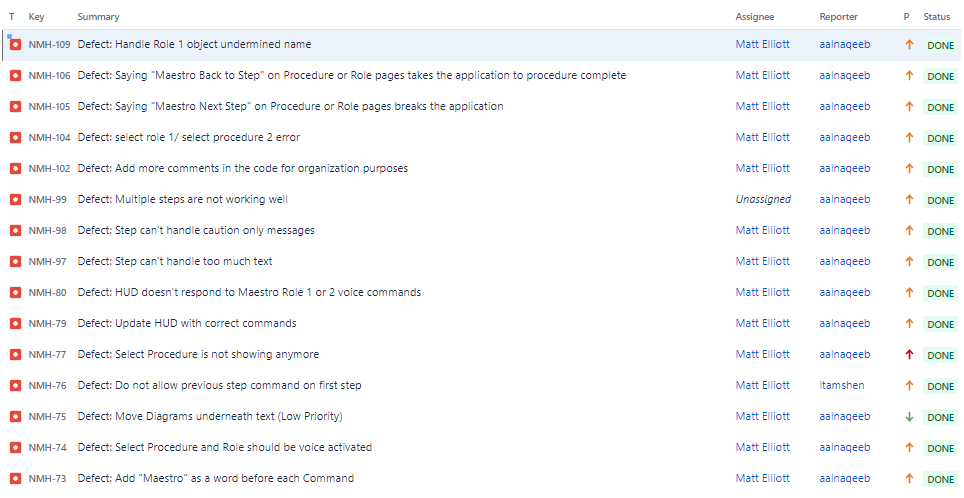
This process pertains to testing the integrated system to ensure that it is still within the scope and the proposed requirements.

### 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.

## Defect Management

Any defects found by the quality assurance team will be submitted through the Jira platform. Below is a list of defects found across all project sprints. These defects were assigned to the developer and were addressed. The QA team retested them and marked them as done upon successful testing.



### 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.

### Defect Assignment

Defects are assigned to the project team lead to be reviewed and assessed. Then the project team lead will reach out to developers to pinpoint resolution and find causes. After these defects are addressed by the developers, the QA team will retest these defects to ensure proper application functionality. All these defects are created off the user story using Jira. Jira platform allows the users to write detailed steps to reproduce the defect. This is important to save the developers time to figure out the issues associated with the defect. Finally, the defect will move to done status after being fixed by the developers and rested by the QA team.

## Project Status

As of 3/23/2020, we lost a team resource which impacted the final deliverable for the NASA Maestro HUD project. As of this writing, it’s likely that the backend code will not connect with the frontend code, ergo, mock data will be included to showcase how the functionality of the project should be. In addition to this, the developer’s guide may not be created for this project. These changes have been noted in the risk assessment table. Kris and James have been notified of this interruption and have approved this impact. Project manager Roy and Dr. Assadullah were also made aware of this change. A summary of details will be outlined in the project closure report document. Project closure report document to be provided at milestone 4. The development environment of the project can be accessed here <https://appdev-nasa-hudweb.herokuapp.com/> and the released version can be accessed here <https://app-nasa-hudweb.herokuapp.com/>

# References

Assadullah, M. (2020). SWEN 670 Capstone Project Guide. Retrieved February 22, 2020, from https://campus.umuc.edu

Montalvo, James. (2020). Project Maestro. Retrieved February 22, 2020, from https://github.com/xOPERATIONS/maestro

Scrum.org (n.d.). The home of scrum. Retrieved March 12, 2020, from https://www.scrum.org