Systems Engineering Management Plan

Hell-ix Group

14th October 2022

Contents

1. Project Structure

1.1 Phase descriptions

The project will be divided in 7 phases plus an exploration phase. The purpose of each of them will be as follows.

1.1.1 Phase 0: Exploration

Investigation of available technologies, team organization and creation of a basic framework of rules for the competition.

1.1.2 Phase 1: Concept of Operations (ConOps) and System Requirements

Further investigations on the technologies to be used and development of documentation detailing system requirements, as well as the user description and summary of the needs, goals, and characteristics of the system's user community.

1.1.3 Phase 2: System High-Level Design

The hardware and digital tool set to be used, as well as the basic structure of the programming will be decided.

1.1.4 Phase 3: System Detail Design and Modeling

Further details about the programming will be defined. The exact hardware to be used will be decided. Appropriate models for simulation will be created and used based on the available data for the chosen hardware.

1.1.5 Phase 4: System Implementation

The drone or its parts will be acquired and assembled if necessary. The necessary code will be created and implemented.

1.1.6 Phase 5: System Testing

Correct functioning of the drone will be tested from the lowest level (regulation of propellers, basic movement, etc) to the highest (AI and ML).

1.1.7 Phase 6: Competition

This phase includes both the joint effort between teams to set up everything necessary for the competition as well as the competition itself.

1.1.8 Phase 7: Retirement

Retirement or disposal of non-reusable materials used in the circuit through the appropriate recycling

1.2 Gantt Diagram

PHASE	START	END
0: Exploration	12-sep	23-oct
1: Concept of Operations (ConOps) and System Requirements	26-sep	11-dic
2: System High-Level Design	07-nov	23-dic
3: System Detail Design and Modeling	05-dic	29-ene
4: System Implementation	30-ene	02-abr
5: System Testing	27-feb	21-may
6: Competition	01-may	04-jun
7: Retirement	05-jun	18-jun

Figure 1.1: Table with starting and ending dates of the project's phases
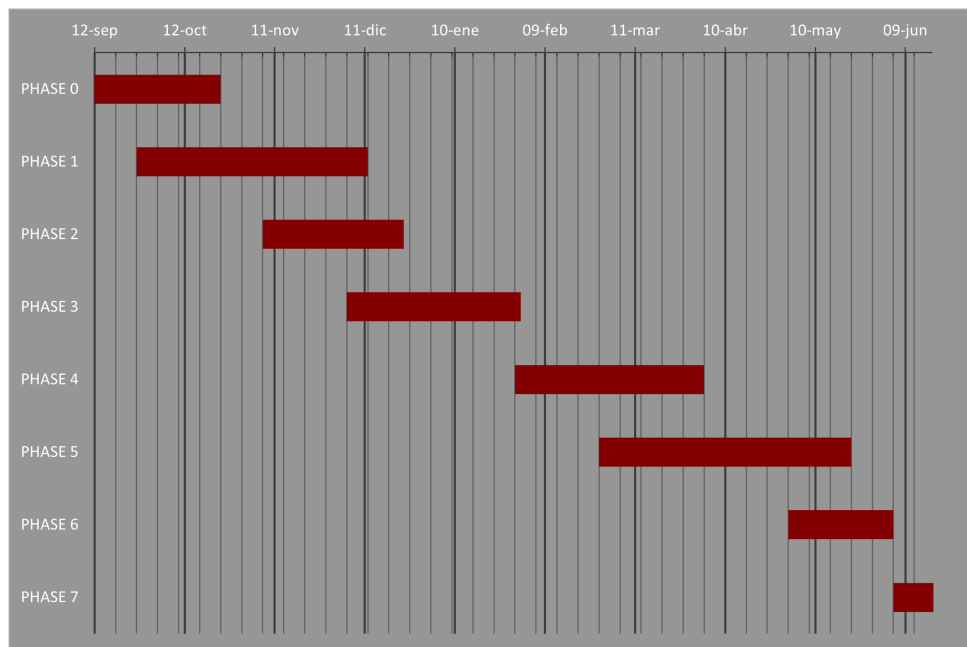

Figure 1.2: Rough Gantt Diagram

2. Engineering Process Control

The purpose of this section is to describe the activities and plans that will act as controls on the project's systems engineering activities. This list of required products will control the activities of the team performing each activity and will control the satisfactory completion of the activity. These plans should be prepared and sharpened when they are clearly needed. Also, other plans might be included when new tasks are undergoing.

2.1 Task Inputs

Every task should be performed according to the standards applying from the following ones:

- ISO/IEC/IEEE Std 15288:2015 Systems Engineering Life-Cycle Processes.

- ISO/IEC/IEEE Std 24748-2:2018 Guidelines for Application of 15288.

- ISO/IEC/IEEE Std 15289:2017 Content of Information Items.

- ISO/IEC/IEEE Std 29148:2018 Requirements engineering.

- OMG SysML Specification v 1.6.

- ISO-IEC Std 14882:2020 Programming languages — C++.

- ISO/IEC 9899:2011 Programming languages — C.

- IEEE 1003.1 POSIX System API Specification.

2.2 Task Deliverables

Every undertaken task should include a report explaining the process followed and its main results. Moreover, each task will have its specific deliverables such as software or hardware produced.

2.3 Task Decision Gates

A list of critical activities that must be satisfactorily completed before a task is considered achieved should be elaborated before start working on the task. Concretely, on the technical side this means:

- Assess that the correct technical metrics were define for the project.

- Assess if the correct technical process was used to carry out the work for the phase.

- Assess the difference between the expected results and the actual results.

On the project management side, this includes:

- Assess the cost for collecting the technical metrics.

- Assess the cost for using the process to carry out the work.

- Assess the difference between the planned baseline cost/schedule and the actual cost/schedule.

2.4   Reviews and Meetings

Each specific team responsible of performing each task will meet to coordinate the different subtasks and also, meet with different teams to align objectives and avoid repetitions. Meetings will be either face-to-face or remote, when necessary, through applications like Microsoft Teams or Zoom.

2.5   Software Development Plan

This plan describes the organization structure, facilities, tools, and processes to be used to produce the project's software. GitHub will be the main platform to develop software tasks, it will be used for both management and communication between different software parts. Also, more platforms could be included for more specific purposes. Software produced will be as simple as possible with a description of every step followed so that anyone can easily understand and modify the code produced. As an example, every number in the code should be named as a variable and include the variable definition.

2.6   Hardware Development Plan

Describes the organization structure, facilities, tools, and processes to be used to produce the project's hardware. Once the drone to be used is defined, hardware development plan will get elaborated. Different alternatives are being studied like buying an existing drone and furnishing it with the necessary tools or buying different components and building a drone from scratch.

2.7   Verification Plan

Its goal is to prove that the system meets the requirements and matches the design. Every deliverable must follow the corporate identity manual, and results obtained match the expectations.

2.8   Installation plan

Describes the sequence in which the parts of the system are installed. Once the different parts, both hardware and software, are defined and developed, an installation plan should be completed to avoid step-backs.

2.9   Configuration Management Plan

Describes the development team's approach and methods to manage the configuration of the system's products and processes. It also describes the change control procedures and management of the system's baselines as they evolve. As a complex project is being approached, it is expected that changes in the system's baselines must be made. Therefore, each task must be clearly defined step by step so that changes can be easily performed.

2.10   Risk management plan

It seeks to understand and avoid the potential cost, schedule, and performance risks to a project. It takes a proactive and well-planned role in anticipating problems and responding to them if they occur. The only certainty is that, in at least some small way, things will not go as planned. Risk management anticipates and controls these risks. Each stage will have its potential risks, so before performing them every step should be analyzed to establish the potential risks and try to minimize them. For example, before buying a drone, all alternatives should be taken into consideration because if an expensive drone is bought and it does not work appropriately, the budget limit will be exceeded.

3. Operational Procedures

In this section, work management will be discussed.

The platform that will be used by every member of the team will be GitHub. This is a website which will help us with version control. There is not a single way of working with this tool but it is really important to have a single and common one for all the team members. Documents uploaded can be project documentation or programming code. They will all be named in the same strict way:

"NameOfTheDocument-Version-Date"

- The name of the document will be written without spaces and with the first letter of each word capitalized. If an acronym for the name exists, it will be used instead and be fully capitalized.

- An underscore will be used instead of a dot in the version number.

- The date will have the format Day_Month.

For example, this document will be uploaded as:

"SEMP-1_0-14_10".

Every document will include a table of updates at the beginning in which it will be specified when who and what was updated. It will look similar to this:

VERSION	AUTHOR	UPDATE	DATE
1.0	Name of the author(s)	New document	XX/XX/XXXX
1.1	Name of the author(s)	Chapter X	XX/XX/XXXX
1.2	Name of the author(s)	Chapter Y	XX/XX/XXXX
2.0	Name of the author(s)	An important change in the document	XX/XX/XXXX

Figure 3.1: Version Table

The manager of each area will be responsible for supervising the correct document naming as well as the contents. Every team member will be aware of what to update, add that information to the Version Table and to follow the format and style of the document. Another assignment for the manager will be supervise the correct document naming as well as the contents.

Every member of the team must know when are the following delivery dates and those ones who have work to do on that have to be well organized to finish it at least one day before deadline. Thus, the manager has time to check and approve that work. Before the last day, feedback is important between both parts, so the last day there is only a few details to check and not the whole document.

To conclude, managers will have to be aware of everything related to the work they lead. They have to be able to make decisions and to be a problem solver for the rest of the teammates. Eitherway, project manager will always have a great weight on the resolution.