**Group Member NAMES: Ryan Sauer, Demetrius Johnson,**

**Jonathan Schall & Olivia Pellegrini**

***Software Quality Assurance***

[**1.0 Introduction 3**](#_uib4xdn0q2uy)

[1.1 Scope and intent of SQA activities 3](#_lpcjwcu19a71)

[1.2 SQA organizational role 3](#_4nmp2raku12d)

[**2.0 SQA Tasks 3**](#_3v5n90m3k10b)

[2.1 Task Overview 3](#_sh3bud49g8s7)

[2.1.1.1 Description of SQA task 1 3](#_qf05da9ks8zl)

[2.1.1.2 Work products and documentation for SQA task 1 3](#_p7gzy4299xov)

[2.1.2.1 Description of SQA task 2 3](#_9y0d7ctnk2yk)

[2.1.2.2 Work products and documentation for SQA task 2 4](#_tn10qgxrh19l)

[2.1.3.1 Description of SQA task 3 4](#_us2d25fzk357)

[2.1.3.2 Work products and documentation for SQA task 3 4](#_bq80yuramzuf)

[2.1.4.1 Description of SQA task 4 4](#_ugvxt2odu6xj)

[2.1.4.2 Work products and documentation for SQA task 4 4](#_oro7ex2f0dt)

[2.2 Standards, Practices and Conventions (SPC) 4](#_syzdbwhwo3fy)

[2.3 SQA Resources 5](#_f2jhmjmc46h2)

[**3.0 Reviews and Audits 5**](#_bymkejh1ii7j)

[3.1 Generic Review Guidelines 5](#_op4nap4qql6z)

[3.1.1 Conducting a Review 5](#_ys4ntnygood2)

[3.1.2 Roles and Responsibilities 5](#_rmfwo7wae16v)

[3.1.3 Review work products 6](#_mhduc5z6r6gb)

[3.2 Formal Technical Reviews 6](#_l25utpamxk1)

[3.2.n Description of review n 6](#_u5foc5u2wshw)

[3.2.n.1 Description and focus of the review 6](#_gz47tb5vo954)

[3.2.n.2 Timing of the review 7](#_dozmj8fgbhok)

[3.2.n.3 Description and focus of the review 8](#_yxmtcrbg4v4b)

[3.2.n.4 Work products produced 8](#_wz66qbsdhp9z)

[3.2.n.5 Review n checklist 9](#_2azkjy5rbbsi)

[3.2.1 System specification review 9](#_mhl63kcbk9z7)

[3.2.2 Software project plan review 10](#_879qwzy8gtnr)

[3.2.3 RMMM review 10](#_kvb5ymmg0bg9)

[3.2.4 Requirements reviews (models, specification) 10](#_ha0e3j9it6lf)

[3.2.5 Data design review 11](#_r4g6v75riu86)

[3.2.6 Architectural design review 11](#_n1d8h2akesd1)

[3.2.7 Interface (GUI) design review 11](#_wtfm28886jt7)

[3.2.8 Component design review(s) 11](#_4u5hx6gp1ymf)

[3.2.9 Code Reviews 12](#_rly7i3bxwiu9)

[3.2.10 Test specification review 12](#_nvoiry6e08g0)

[3.2.11 Change control reviews and audits 12](#_x0pgb75y3fvn)

[3.3 SQA Audits 12](#_eozb0vs45hjo)

[**4.0 Problem Reporting and Corrective Action/Follow-up 12**](#_wsj1xsfyy50e)

[4.1 Reporting mechanisms 13](#_rnos0b4onfth)

[4.2 Responsibilities 13](#_w72brof2w4n1)

[4.3 Data collection and evaluation 13](#_ye6jrsjck3i1)

[4.4 Statistical SQA 13](#_o99nkux5uf1c)

[**5.0 Software Process Improvement Activities 14**](#_4xd3ahlf9xwa)

[5.1 Goal and objectives of SPI 14](#_46hei7765q3n)

[5.2 SPI tasks and responsibilities 14](#_21gncbkd2bi)

[**6.0 Software Configuration Management Overview 15**](#_gijjqcmx8we2)

[**7.0 SQA Tools, Techniques, Methods 15**](#_zd1rovgt79ny)

[**8.0 Appendix 15**](#_oqm65ntdyiz5)

# 1.0 Introduction

## 1.1 Scope and intent of SQA activities

The objective of the SQA is to ensure that the team does not deviate from the plan and requirements set by the client. This will ensure that the team accomplishes tasks according to the schedule and properly follow the plan unless otherwise specified. Any deviation from the plan without prior approval will not be tolerated and will be discussed with the client in most cases.

## 1.2 SQA organizational role

The SQA team is the same as the development team, as we are only four people and do not have any other assistance. All members of the team will share responsibilities, however some roles will mainly focus on certain aspects, such as the vehicle administrator being responsible for the majority of testing.

# 2.0 SQA Tasks

## 2.1 Task Overview

### 2.1.1.1 Description of SQA task 1

The software developers will meet bi-weekly with the entire team and client and give a rundown of changes, progress, and demonstration (if applicable) of the current code base to ensure that the previous two week's work aligns with the Project Plan and continues to perform as expected. We are using in person meetings or the Zoom meeting application.

### 2.1.1.2 Work products and documentation for SQA task 1

A semester-long document will hold meeting notes that hold the feedback given by the team and client that outlines the future goals and objectives for the software developers to follow along with the project plan for the future weeks.

### 2.1.2.1 Description of SQA task 2

The network architect will meet bi-weekly with the entire team and client and give a rundown of plans, research, and decisions to ensure that they are compatible with the software engineer code, suitable for the client, and effective in practice.

### 2.1.2.2 Work products and documentation for SQA task 2

A semester-long document will hold meeting notes that hold the feedback given by the team and client that outlines the direction and specifications for the network architect to follow along with the project plan for the future weeks.

### 2.1.3.1 Description of SQA task 3

The vehicle administrator will meet bi-weekly with the entire team and client and give a rundown of the changes to the hardware, implementation of the software, and to describe test results to ensure they are maintaining the needed materials and performing the correct tests. Changes should be in the scope of what the team has agreed upon, and any changes outside of that must be approved before committed.

### 2.1.3.2 Work products and documentation for SQA task 3

A semester-long document will hold meeting notes that hold the feedback given by the team and client that outlines the list of code and programs to implement along with adjustments to the hardware that need to be made for the vehicle administrator to follow.

### 2.1.4.1 Description of SQA task 4

The appointed project manager will ensure that bi-weekly meetings are scheduled, documented, and ensure the dissemination of information is guaranteed for those who miss meetings or need more frequent meetings. Such meetings include 1-on-1s with the network engineer and vehicle administrator, network engineer and software engineers, and vehicle administrator and software engineers.

### 2.1.4.2 Work products and documentation for SQA task 4

All meetings, including the bi-weekly one should be kept on documents in a shared drive in a labeled folder for meeting notes. These will be kept up to date and brought to all team members' attention when updated by the project manager via communication channels (such as Google Chat or email).

### 2.2 Standards, Practices and Conventions (SPC)

There will be bi-weekly meetings with every party involved to ensure the project is on trajectory to be completed on schedule with accuracy. This meeting will involve every team member doing a verbal presentation of the work they have completed over the past two weeks along with doing verbal or visual demonstrations of the work’s current function. The client and other team members will then give verbal feedback that will be documented for reference.

There will also be unscheduled meetings periodically as needed between roles to ensure that both parties are working together and achieve the shared function. These meetings will be documented just as the bi-weekly meetings are for the rest of the team to catch up and double check. These meetings will also be mentioned during the bi-weekly meetings.

Any large scale changes or adjustments must be brought to the team during the bi-weekly meetings and will be decided on democratically. Should there be a tie (the team has 4 members) or the change would impact the product’s final functionality, the client's vote will be taken into account as a tiebreaker or veto.

All major defects should be reported as soon as possible to the project manager. Adjustments will be made by the project manager or the team as a whole depending on the severity of the situation as decided by the project manager.

### 2.3 SQA Resources

The shared drive for all documents has been shared with all team members and the client using Google Drive. The team is also using a Github repository to store all project information and make code updates. The Google Drive is used mostly for documentation, and the Github repository is used mostly for code and other software files used to implement the project; although, the Google Drive folder is used as another backup source for software and code files as well for redundancy. Another team member is also backing up all project-related data files on another home private server for another layer of data redundancy. All software and documentation files used for this project are outlined in the Appendix of the Software Requirements Specification Document (Here: [SRS](https://docs.google.com/document/d/1CzLI-R7gJvZXEFBdx_cmSvVn-LrMk3Is7ne4HMdQKmA/edit?usp=sharing)).

# 3.0 Reviews and Audits

*This section discusses major project reviews conducted by SQA staff and software team members*

## 3.1 Generic Review Guidelines

*A set of guidelines for all formal technical reviews (FTR's) is presented in this section.*

### 3.1.1 Conducting a Review

*General guidelines for conducting a review.*

A formal technical review should use these general guidelines:

i. Should follow a schedule to conduct reviews regularly

ii. Limit the numbers of reviewers

iii. A reviewer needs to keep note of issues that have been addressed

iv. Review the product to ensure it meets requirements and looks for errors

V. Reviewers should receive material before meeting and be prepared to discuss their reviews.

### 3.1.2 Roles and Responsibilities

*The roles people play during a FTR and the responsibilities of each player*.

* Moderator (Ryan) - Review leader, in charge of leading discussions and scheduling meetings
* Scribe (Demetrius) - Keeps note of errors found during FTR
* Reviewers (Jonathon, Demetrius, Olivia, Ryan)- Inspect material for defect, usually prior to the meeting
* Manager(s) (Professor Song)- Has the final say on execution of reviews

### 3.1.3 Review work products

*Documents, forms, lists produced as a consequence of the FTR*.

After reviewing the software, attendees should approve of the decision that was chosen. Documents produced as a result of the FTR are:

* Review issue list
* Review summary
* Updates to schedule, if needed

## 3.2 Formal Technical Reviews

*A description of the specific character and intent of each major FTR conducted during the software process*.

### **3.2.n Description of review *n***

The sections that follow are included for each Section 3.2.n

[1] Introduction of Project and Client

[2] Review project specifications ([SRS](https://docs.google.com/document/d/1CzLI-R7gJvZXEFBdx_cmSvVn-LrMk3Is7ne4HMdQKmA/edit?usp=sharing)) document pt. 1

[3] Review [RMMM](https://docs.google.com/document/d/1eSKZZmf55wZ8s7j0sWaZZmcPMFLtaNlcUYi8I8vuBi4/edit?usp=sharing) document

[4] Review final [SRS](https://docs.google.com/document/d/1CzLI-R7gJvZXEFBdx_cmSvVn-LrMk3Is7ne4HMdQKmA/edit?usp=sharing) document pt. 2 and present the working prototype to client

[5] Present Prototype to Client and make sure we meet requirements of SD1\*

[6] Review schedule for SD2\* and make sure we are on track

\*SD stands for Senior Design course being held at University of Michigan-Dearborn (Jan-August, 2023)

### 3.2.n.1 Description and focus of the review

The overall goals of FTR include:

* Uncover errors, logic or implementation in software
* Ensure the software meets the requirements
* Check that software follows the predefined standards
* Develop software that in a uniform matter
* Make our project manageable

Our FTRs focused on:

[1] Review for Project Requirements such as timelines, roles, and software specifications. Review the backlog, and Gantt chart, as well as getting it approved by the client. We found roles according to the skills of our team members and discussed if the timeline was realistic.

[2] In this review, we went over the SRS document. We discussed the goals, constraints, and use cases within our project. We decided the three use cases in our project are running an experiment, changing car and drone parameters, and retrieving data.

[3] The RMMM document review helped us revise the risks in our project. We discussed what is most likely to happen. We discussed possible solutions. For example, getting a back-up drone.

[4] For the pt. 2 of the SRS we created activity diagrams and state transition diagrams to show how our software behaves. We also addressed the restrictions, limitations, and constraints of our project.

[5] In this meeting, we discussed with the client what the working prototype is as we near the end of SDI. We decided that having the drone flying and detecting an object, and the car driving should be our prototype.

[6]In this meeting, we discussed what work we have left in our project as the semester comes to an end. We looked at the schedule we created to see if it is still accurate and discussed if any requirements changed.

### **3.2.n.2 Timing of the review**

Our team meets bi-weekly regularly with our client for a Formal Technical Review. In these meetings, the team members come with concerns about our work that we have completed since the last meeting. We do this to discuss any concerns or questions we want cleared by our client before continuing to develop on top of our already existing software.

[1] Feb 4 - This meeting took place at the beginning of our class. This allowed us to divide up tasks for members before starting to work on the project.

[2] Feb 18 - For this meeting we first needed the pt 1 of the SRS document to be completed and reviewed by the team members.

[3] March 4 - We reviewed the RMMM document. We had to have this meeting after the RMMM document was completed.

[4] March 18 - We had to finish pt 2 of the SRS document before holding this FTR.

[5] April 1 - We needed to have this meeting to make sure we have provided a sufficient prototype. We also wanted to do this before the Senior Design Day competition to provide an image of our experiment running on the handout.

[6] April 16 - In this meeting, we looked over the schedule we created at the beginning of the class to make sure we had everything done before the class came to an end. We still wanted enough time to complete unfinished tasks if needed.

### **3.2.n.3 Description and focus of the review**

The goals of FTR include:

* Uncover errors, logic or implementation in software
* Ensure the software meets the requirements
* Check that software follows the predefined standards

[1] Review for Project Requirements such as timelines, roles, and software specifications. Review the backlog, and Gantt chart, as well as getting it approved by the client. We found roles according to the skills of our team members and discussed if the timeline was realistic.

[2] In this review, we went over the SRS document. We discussed the goals, constraints, and use cases within our project. We decided the three use cases in our project are running an experiment, changing car and drone parameters, and retrieving data.

[3] The RMMM document review helped us revise the risks in our project. We discussed what is most likely to happen. We discussed possible solutions. For example, getting a back-up drone.

[4] For the pt. 2 of the SRS we created activity diagrams and state transition diagrams to show how our software behaves. We also addressed the restrictions, limitations, and constraints of our project.

[5] In this meeting, we discussed with the client what the working prototype is as we near the end of SDI. We decided that having the drone flying and detecting an object, and the car driving should be our prototype.

[6]In this meeting, we discussed what work we have left in our project as the semester comes to an end. We looked at the schedule we created to see if it is still accurate and discussed if any requirements changed.

### **3.2.n.4 Work products produced**

We were able to collect notes from each FTR and receive advice from our client. We recorded notes on any errors that needed to be fixed or requirements/expectations that still need to be met. We create folders and put our notes in the folder for all team members to view. (error log file: <https://docs.google.com/spreadsheets/d/1Ebp238diHMRFj3LD2SD7EBzl8BeOel0Eer0fs3e_PBU/edit?usp=share_link>)

[1] Gnatt Chart and Schedule for tasks

[2] Finalized SRS document pt. 1, notes from meeting with client

[3] Finalized RMMM document, notes from meeting with client

[4] Finalized SRS document pt. 2, notes from meeting with client

[5] Video of prototype, video of object detection, notes of topics discussed

[6] Update Gnatt chart and schedule, notes from meeting

### **3.2.n.5 Review n checklist**

*The sections that follow represent typical reviews conducted as part of a software engineering project and are included as part of Section 3.2.n*

[1] Introduction of Project and Client:

* Are the roles on our team allocated evenly?
* Are we utilizing each member’s strengths?
* Do we have a team leader?
* Do we have a written schedule and is it approved by the client?
* Are the objectives we outlined achievable within our timeline?

[2] Review project specifications ([SRS](https://docs.google.com/document/d/1LsXinBeIe5RTQ6oee1FFdQJKwKT-5xNaWBrpxOrkHEg/edit?usp=sharing)) document pt. 1:

* Are all use cases covered in this document?
* Have we described all constraints in our document?
* Have we clearly defined our users?
* Have we defined the software context? (i.e. for research purposes)
* Do we have sequence diagrams to describe scenarios within our system?

[3] Review [RMMM](https://docs.google.com/document/d/1eSKZZmf55wZ8s7j0sWaZZmcPMFLtaNlcUYi8I8vuBi4/edit?usp=sharing) document

* Are our risks outlined in our document realistic?
* Have we figured out what our greatest risks are?
* Are there ways to manage these risks described?
* Are all risks covered within this document?

[4] Review final [SRS](https://docs.google.com/document/d/1LsXinBeIe5RTQ6oee1FFdQJKwKT-5xNaWBrpxOrkHEg/edit?usp=sharing) document pt. 2 and present the working prototype to client

* Are our data descriptions clearly stated?
* Do we have a UML diagram to represent our system?
* Do our state and activity diagrams accurately depict our system?

[5] Review schedule for SD2 and make sure we are on track

* Have we completed all tasks assigned in SDI?
* Does our client approve of our schedule for SDII?
* Do we have time to detect more than just a red object?

### 3.2.1 System specification review

[Here](https://docs.google.com/document/d/1LsXinBeIe5RTQ6oee1FFdQJKwKT-5xNaWBrpxOrkHEg/edit?usp=sharing) is our Software Requirement Specification (SRS) document, which discusses objectives, use cases, and description of software behavior.

After our meetings, we decide if any of our system specifications need to be altered. Thus far our goals and objectives have remained the same. The client has not changed his ideas about what is expected out of this project for the most part. He does want to see how much we achieve by the end of Senior Design, and may request more objects to be detected depending on if we have time.

### 3.2.2 Software project plan review

[Here](https://docs.google.com/document/d/1LsXinBeIe5RTQ6oee1FFdQJKwKT-5xNaWBrpxOrkHEg/edit?usp=sharing) is our Project Plan document we provided early.

This is a review of the overall project which includes a timeline and outline of tasks. Our team has an Excel sheet and uses Jira to help us keep track of our timeline and distribution of tasks. (Link to schedule: <https://docs.google.com/spreadsheets/d/1Np0KL0MUnTC7GoU5v01r7UdblBakfGlUsDKDI06vU_Y/edit#gid=0>) .

A software project plan review can help us see if we are still on track with the original schedule or if we need to adjust the dates. In this review, we can see if our project is still within budget, if roles/responsibilities are still clearly defined, and if we have access to resources needed to complete the project.

For our project, our team met with the client early into our project to distribute responsibilities of each person. We also met to discuss the resources we need, such as RPis, a drone, and a PiCarX. Recently, we discussed with our client getting a back-up Clover drone in case something happens to the original drone, and checked if it was within budget. As we complete specific tasks, these reviews help us to continuously hand out tasks to team members once older tasks are completed.

### 3.2.3 RMMM review

[Here](https://docs.google.com/document/d/1eSKZZmf55wZ8s7j0sWaZZmcPMFLtaNlcUYi8I8vuBi4/edit?usp=sharing) is our Risk Mitigation, Monitoring and Management (RMMM) document.

In our meetings with our client we do address the risks in our project. As stated in the previous section, after testing we found that our equipment risk is still one of the most prominent risks. This is why we have brought up the idea of getting another drone to work with. So far that is the only risk we have felt is a concern affecting our team.

### 3.2.4 Requirements reviews (models, specification)

[Here](https://docs.google.com/document/d/1CzLI-R7gJvZXEFBdx_cmSvVn-LrMk3Is7ne4HMdQKmA/edit?usp=sharing) is our Software Requirement Specification (SRS) document, which discusses objectives, use cases, and description of software behavior.

For the most part, our requirements about the models and specifications have remained the same throughout our project. The client has not expressed any new expectations of us yet. As we get further into the second half of Senior Design, we will be expected to detect more objects than just red vehicles. We will have to discuss later with our client what he wants us to detect to make our project more applicable to real world use. Likely, we would need to train a model with vision data of accidents and various vehicle types so it can make more real-world, comprehensive detections.

### 3.2.5 Data design review

Data design review addresses the data flow between each component (drone, car, software) within our project.

The data flow begins when the user inputs parameters initiating the experiment to run. The car will receive this and begin moving. As the car moves, the drone will follow above it to begin detection. Once detection occurs, the drone will tell the car to stop.

As we are currently testing our experiment, we have only begun collecting data about the network bandwidth between the drone and the car; we still need to acquire and decide on an interface to gather other measurements such as battery status, and latency. We plan to do this in the future, as it is one of our main objectives.

So far, we have been recording videos from the drone’s camera of it detecting a red object and flying. We soon hope to collect data about the drone and car communications so it is viewable by the user in some graphical format. This information can be kept in a file and analyzed later to test the efficiency of our drone-car collaboration.

Overall, the final goal for the design of the data flow remains the same, though it does change as we test our experiment. The final product will stick to the initial design.

### 3.2.6 Architectural design review

For an architectural design review, our team reviews the project, design, layout, and data flow. Our team has decided with the clients permission that to perform our experiment in the early stages, we can fly the drone over the car using the controls rather than following the car on its own. We will do this by using Aruco Map Markers so that the drone has a local map and a path that it can follow that will be the same course as the car. Of course as we work on the project more, we plan to have the drone follow the car, where the car is itself the marker or mechanism that controls the drone’s path.

For initial testing, we plan to only use ArUco markers to set the trajectory of the drone’s flight. We will hopefully not need this for route planning in the future since the goal is to have the drone follow the car.

The final design will remain as we envisioned in the beginning of our project, but it will change as we test and run our experiments.

### 3.2.7 Interface (GUI) design review

Our project interface is only used by team members. We input parameters to a terminal to run this project. The average person is not going to be using our work since it is more research based, unlike something like a mobile application or website. So no changes need to be made to our interface. However, if we have enough time, we can easily design a simple GUI interface to make it easier to run experiments and receive output messages. So far the only GUI we have is the web interface that is part of the Clover Drone software Robot Operating Systems topics (ROS Topics) used to post information from drone sensors.

### 3.2.8 Component design review(s)

Beside an object detection program, we plan to collect data from the drone about battery life, bandwidth, and latency. In the future, we plan to detect more specific objects, rather than just a red object, to make our project more useful to real world scenarios.

### 3.2.9 Code Reviews

Our team uses code reviews to ensure the quality of our software. We performed code reviews on our object detection program using OpenCV-Python. We first used pair programming to develop our program. Then, we conducted a FTR, which included the rest of our team and client, to finalize our program before implementing it on the drone.

### 3.2.10 Test specification review

Test specifications are the way we plan to test our experiment. Since reviewing this with our client, what we test will likely change. We have been using the red side of a Rubik's cube to test our program, but plan to use another car to make our test more realistic. We need to test our program in different environments to make sure that our program can detect objects in different lightings, altitudes, and speeds.

### 3.2.11 Change control reviews and audits

Our team addresses issues that arise in the project and the changes we want to make as a group. We then present the desired changes to the client in our bi-weekly meetings to be sure we have his approval before actually implementing the change. We then determine if these changes helped improve any weakness there were prior in the project. We are sure to take note of any side effects of changes in our project.

## **3.3 SQA Audits**

*A description of audits performed by the SQA group with the intent of assessing how well SQA and software engineering activities are being conducted on a project*.

To determine if there are any weaknesses occurring on our team/project, Each week on Sunday our team lead checks in with us to assess where we are work-wise individually, and addresses any concerns we have. We are reminded of what we need to be doing to stay on schedule.

Each Wednesday after class we assess what we need to improve on and work quicker on to meet deadlines we created.

Concerns are also brought up in the bi-weekly meeting with our client if we cannot resolve it ourselves.

# 4.0 Problem Reporting and Corrective Action/Follow-up

This section describes problem reporting mechanisms that occur as a consequence of the FTR's that are conducted and the means for corrective action and follow-up.

## 4.1 Reporting mechanisms

We will use a Google Sheet in the shared drive to log errors and problems discovered during testing. This is a convenient and centralized method that every member of the team including the client is able to access and use. If a higher level issue is identified that needs immediate attention, the direct forms of communication such as email or a messaging app should be used to get the proper experts attention.

## 4.2 Responsibilities

Our team is small enough that larger decisions will be made democratically. Ryan has been designated as the project manager and will be responsible for ensuring everyone is keeping up with their responsibilities. In the event of a smaller issue, the person in charge of the related design will be responsible for ensuring it is resolved. For example, should an algorithmic issue come up, that would be on a software developer. The roles are as follows:

* Network Architect and Administrator: Ryan Sauer
* Software Developer: Olivia Pellegrini & Jonathan Schall
* Vehicle Administrator and Developer: Demetrius Johnson

## 4.3 Data collection and evaluation

Data produced by the system will be collected by the Vehicle Administrator. However, data collection design will be created by the Network Architect along with data evaluation.

Data about the process will be collected by the project manager for evaluation and then retransmitted to all necessary parties. This will be done at every bi-weekly meeting with the client and team.

## 4.4 Statistical SQA

All problems reported to the project manager will be logged, identified, and analyzed for future improvement. Every issue will be sorted into one of the following problems:

* Incomplete specification
* Misinterpretation of client needs
* Intentional deviation from specification
* Violation of programming standards
* Error in data design
* Incomplete testing
* Inaccurate documentation
* Error in programming language translation of specifications
* Inconsistent human-computer interface
* Other

If a trend is found in terms of the quantity or a recurrence of a specific problem, the project manager will work with the team to ensure a resolution for the source of these issues. The further the project moves forward, the smaller the threshold for project manager intervention will be.

# 5.0 Software Process Improvement Activities

The SQA group (and others) is often chartered with responsibility for software process improvement (SPI). This section describes the work associated with SPI.

## 5.1 Goal and objectives of SPI

The goals of SPI are to reduce the number of errors that appear during development and to avoid common mistakes made early on in development. To do this we will properly document issues found and what causes them, along with their impact on the schedule. We will also refer to documentation for the drone and the car, found on their respective websites and Github Repositories where there are also documented known issues.

## 5.2 SPI tasks and responsibilities

Errors are recorded in a table with a name, type, description, and cost. Common errors and/or errors with high cost will be prioritized as the main errors to determine causes for and to eliminate from the project.

Below are the main error types we expect to encounter during development.

| Error Type | Description |
| --- | --- |
| Hardware Configuration | Both the PiCarX and the Clover Drone need to be properly set up to maneuver, communicate, read and store data, and use various sensors (especially the drone camera). Early in development we will likely run into issues with getting all of the hardware to work |
| CV Program | Our work relies heavily on complex computer vision programs to detect objects and collect position information of the drone. We are also very inexperienced with cv programming and will likely make mistakes and need to continuously make corrections. |
| Drone Flight Control | Autonomously controlling the drone requires precision and even with simple pathing will take time to learn how to do properly |
| Other | We will likely run into a few other errors regarding communication, processing, data collection and storage, etc that don’t fit into these categories |

# 6.0 Software Configuration Management Overview

When a change needs to be made to the original plan the team needs to identify the change and the reason it is necessary to make the adjustment. Then, the negatives to making the change must also be identified. If the positives outweigh the negatives, the change should be presented to the client for approval. If approved, the schedule and documentation should be updated to reflect the changes.

# 7.0 SQA Tools, Techniques, Methods

SQA meetings will be held at least once every other week. All members need to attend an SQA meeting at least once every other week. If a member does not attend for two or more meetings in a row the team and the client will be notified.

All documentation and software created both in and out of the SQA meetings relating to the project is required to be in a shared drive accessible to the team and the client.

All scheduling is done using Jira. All members of the team are assigned tasks and are notified of any changes to their tasks. As mentioned in section 6, any changes to the plan needs to be reflected in the schedule.

# 8.0 Appendix

Supplementary information is provided here.

* S. A. Hadiwardoyo, E. Hernández-Orallo, C. T. Calafate, J. -C. Cano and P. Manzoni, **"Evaluating UAV-to-Car Communications Performance: Testbed Experiments,"** 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), Krakow, Poland, 2018, pp. 86-92, doi: 10.1109/AINA.2018.00025.
  + **URL:** <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8432227&isnumber=8432202>
* J. Yoon, I. Kim, W. Chung and D. Kim, **"Fast and accurate car detection in drone-view,"** 2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Seoul, Korea (South), 2016, pp. 1-3, doi: 10.1109/ICCE-Asia.2016.7804775.
  + **URL:** <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7804775&isnumber=7804716>
* Yildiz, Melih, Burcu Bilgiç, Utku Kale, and Dániel Rohács. 2021. **"Experimental Investigation of Communication Performance of Drones Used for Autonomous Car Track Tests"** Sustainability 13, no. 10: 5602.
  + **URL:**

<https://doi.org/10.3390/su13105602>

* Barbeau, Michel, Joaquin Garcia-Alfaro, and Evangelos Kranakis. 2022. **"Research Trends in Collaborative Drones"** Sensors 22, no. 9: 3321.
  + **URL:**

<https://doi.org/10.3390/s22093321>

* Textbook: **Software Engineering A PRACTITIONER'S APPROACH EIGHTH EDITION, by Roger S. Pressman, Ph.D. and Bruce R. Maxim, Ph.D.**
* Scheduling software service we are using:
  + <https://www.atlassian.com/software/jira>
* Video conference software we are using for meetings:
  + <https://zoom.us/>
* Shared database software we are using (Google Drive and Github):
  + <https://www.google.com/drive/>
  + <https://github.com/meechtheballer99/Drone-Car-Collaboration>
* Shared Error log file:
  + <https://docs.google.com/spreadsheets/d/1Ebp238diHMRFj3LD2SD7EBzl8BeOel0Eer0fs3e_PBU/edit?usp=share_link>