SE 4485: Software Engineering Projects

Fall 2025

Project Management Plan

|  |  |
| --- | --- |
| Group Number | Team 1 |
| Project Title | City Level Air Quality Prediction Application |
| Sponsoring Company | Raytheon (Team A) |
| Sponsor(s) | Ryan Havens <Ryan.Havens@rtx.com>,  Marc Perna <marc.perna@rtx.com>,  Trey Williams <trey.williams@rtx.com>,  Trevor Lang <trevor.a.lang@rtx.com> |
| Students | 1. Jay Chung <cwc130330@utdallas.edu>  2. Amelia Quinn <qcb220000>  3. Kevin Melo <ksm220005>  4. AJ Kimbrough <ank210005>  5. David Santos <des210001>  6. Andrew Einright <ame210008> |

**ABSTRACT**

This document defines the project organization, lifecycle model, risk analysis, required resources, scheduled deliverables, professional guidelines, and configuration management for the City Level Air Quality Prediction (CLAP) Application.

# **TABLE OF CONTENTS**

1. Introduction
2. Project Organization
3. Lifecycle Model Used
4. Risk Analysis
5. Software And Hardware Resource Requirements
6. Deliverables and Schedule
7. Monitoring, Reporting, and Controlling Mechanisms
8. Professional Standards
9. Evidence of Configuration Management

# 

# **LIST OF FIGURES**

TBD

# 

# **LIST OF TABLES**

* + - 1. Table 1 – Includes a brief risk analysis, including details of the following: risk likelihood, impact, mitigation strategies, and rationale
      2. Table 2 – Includes information regarding project milestones, due date, allocation of responsibility, dependencies, and the estimated time of completion.
      3. Configuration Management Log – Tracks document/code revisions that were made using GitHub

**INTRODUCTION**

This document provides a project plan for the development of the City Level Air Quality Prediction (CLAP) Application. The purpose of the document is to outline how CLAP will be designed, developed and deployed. The scope of the document is to define the structure, design, and management for the development of the CLAP system. The CLAP system is a predictive analytics application designed to predict future AQI category for a single U.S. city using historical AQI and weather data. CLAP can be utilized as an educational tool for students interested in building similar projects. Depending on the quality of data provided, CLAP will be capable of anticipating future weather trends, which may provide actionable insights for users. This document is organized as follows: project organization, lifecycle, risk analysis, tools, deliverables, project management, professional standards and configuration management.

**PROJECT ORGANIZATION**

Team 1 (or Raytheon Group A) currently consists of 6 software engineers assigned to a single group. This assignment is a temporary arrangement until the project workload can be properly assessed. The rationale for this arrangement is that the number of modules is unknown until proper requirements for this project have been determined. When the team reasonably understands the project’s architecture and design, they will then be able to divide the workload by separating the current team into multiple groups. As of now, the plan will be to undergo continuous iterative development until the project’s requirements and specifications are more intuitive. I believe that working as one group at this moment will encourage communication, ensuring that everyone is on the same page in the early phases of the project.

Team Members and Roles:

1. Jay Chung (cwc130330) - Group 1 Team leader, Software & AI Engineer
2. Amelia Quinn (qcb220000) - Software & AI Engineer
3. AJ Kimbrough (ank210005) - Group 1 Lead Architect, Software & AI Engineer
4. Kevin Melo (ksm220005) - Software & AI Engineer
5. David Santos (des210001) - Software & AI Engineer
6. Andrew Einright (ame210008) - Software & AI Engineer

# **LIFECYCLE MODEL USED**

Our team has chosen an iterative lifecycle model to guide the development of the project, as it allows us to refine the system through repeated cycles of feedback and improvement. This approach is good for demonstrating the project as a proof of concept, since it enables early validation while progressively enhancing non-functional requirements.

# **RISK ANALYSIS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk:** | **Likelihood:** | **Impact:** | **Mitigation:** | **Rationale:** |
| Implementation Complexity | Less than 50% | More than 50% | Simplify the requirements. | If the project scope expands beyond requirements, then the project will become more complex, making it difficult to complete the project on time. |
| AQI Data Inconsistency | Less than 50% | More than 50% | TBD | If historical AQI data is inconsistent or unreliable, then the model’s predictions will be less accurate, resulting in a bad product. |
| API Unavailability | Less than 25% | More than 50% | TBD | If the AQI Data API becomes unavailable, then the system will be unable to retrieve real-time data, limiting functionality. |
| Lack of Team Coordination | Less than 50% | Less than 50% | Weekly Meetings, clear task ownership | If the project team does not maintain communication and task ownership, then development process will slow and errors may increase. |

# Table 1. Risk Analysis – Details the risk’s likelihood, impact, mitigation strategies, and rationale.

# **SOFTWARE AND HARDWARE RESOURCE REQUIREMENTS**

We are currently exploring our software options until specific requirements are set in place. We believe that the key to successfully completing this project is to keep things simple. We plan on developing, testing and deploying the project application using software downloaded on our student laptops. To keep things simple, the database will likely require one or two tables for holding AQI data. Cloud service may not even be required. We will be using GitHub for configuration management for simplicity. We are considering creating a video of our demonstration as a contingency measure.

Software:

1. Python – Many of our members are proficient at utilizing this language.
2. SQLite database – Some of our members are proficient at utilizing this database.
3. GitHub for CI/CD – For simplicity and ease of use.

Hardware:

1. Student laptop – For simplicity and ease of use.

# **DELIVERABLES AND SCHEDULE**

This section includes information regarding important milestones and their deadlines. The estimated time of future milestones (or deliverable) will be determined upon the completion of its dependent milestone.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Deliverable:** | **Due Date:** | **Allocation of Responsibility** | **Dependencies:** | **Estimated Time:** |
| Weekly Report | Every Friday | Team leader | N/A | >0.5 hour |
| Project Management Plan | 09/12/2025 | Entire Team | Must be approved by sponsors. | 2 hrs |
| Requirements Documentation | 09/26/2025 | Entire Team | Must be completed before Architectural Design. | TBD |
| Architecture Documentation | 10/24/2025 | Entire Team | Must be completed before Detailed Design. | TBD |
| Detailed Design Documentation | 11/07/2025 | TBD | Must be completed before Testing. | TBD |
| Test Plan | 11/21/2025 | TBD | Must be considered during Requirements Specification and Architectural Design. | TBD |
| Final Project Presentation Slides | 12/02/2025 | TBD | TBD | TBD |
| Final Project Report | 12/05/2025 | TBD | TBD | TBD |

# Table 2. Deliverables and Schedule – Details information regarding delivery milestones, including the name of the milestone, due date, allocation of responsibility, dependencies, and estimated time.

# **MONITORING, REPORTING, AND CONTROLLING MECHANISMS**

* Weekly Attendance Reports must be produced based on Weekly Progress Meetings with the sponsors, to be submitted every Friday of that week.
* Weekly Sponsor Reports are recommended at least once every week.
* GitHub is recommended for version control and configuration management.
* Weekly Status Reports are recommended for scheduling and meeting important deadlines.

# **PROFESSIONAL STANDARDS**

* Academic integrity
* Respect for all team members
* Equal distribution of workload
* Timely delivery of assigned tasks
* Good behavior (e.g. not missing deadlines and not submitting poor quality work)

**EVIDENCE THE DOCUMENT HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT**

The name of the configuration tool is GitHub. The table below tracks information regarding version control.

https://github.com/cchung7/rtx\_team1/blob/main/group1-Project%20Management%20Plan.docx

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date of Change:** | **Version Before:** | **Version After:** | **Author:** | **Difference (Link):** | **Review -Change Summary:** | **Reviewers:** |
| 9/10/2025 | v0.1 (3cd1923) | v0.2 () | cwc130330 |  | Necessary corrections made to all sections. | Amelia Quinn |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Table 3. Configuration Management Log – Tracks document/code revisions that were made using GitHub.

**ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS**

* IEEE Std 1058-1998: Software Project Management Plans [[pdf](https://course.techconf.org/se4485/IEEE/IEEE-Std-1058-1998-Software-Project-Management-Plans.pdf)]
* PMBOK® Guide: Project Management Body of Knowledge [[pdf](https://course.techconf.org/se4485/IEEE/PMBOKR.pdf)]
* IEEE Std 12207: Software Life Cycle Processes [[pdf](https://course.techconf.org/se4485/IEEE/IEEE%2012207%20(2017)%20-%20Software%20Life%20Cycle%20Processes.pdf)]
* IEEE Std 15939: Measurement Process [[pdf](https://course.techconf.org/se4485/IEEE/IEEE%2015939%20(2017)%20-%20Measurement%20Process.pdf)]
* ISO/IEC/IEEE Std 29148-2018: Systems and Software Engineering

§ Life Cycle Processes

§ Requirements Engineering [[pdf](https://course.techconf.org/se4485/IEEE/ISO-IEC-IEEE-29148-2018.pdf)]

**ADDITIONAL REFERENCES**

* Larson, E. and Gray, C., 2014. Project Management: The Managerial Process. McGraw Hill
* Humphrey, W.S. and Thomas, W.R., 2010. Reflections on Management: How to Manage Your Software Projects, Your Teams, Your Boss, and Yourself. Pearson Education

**Appendix A.**

The following provides a professional standards guideline for the teams. This guideline may be tailored.

Guideline:

On the first occurrence of unacceptable behavior, determine the circumstances involved, resolve the problem, and document the event in the meeting minutes.

On a second occurrence, notify the instructor of the problem. A meeting will be set up to evaluate the situation and resolve the problem.

On a third occurrence, again notify the instructor of the problem. A meeting will be set up to evaluate the situation and resolve the problem. At this point, the team will have the *option* of removing the team member. If removed, then the team member receives a pro-rated grade based on the number of weeks they have participated in the group.

Examples of unacceptable behavior may include not delivering on time, delivering poor quality work, missing team meetings, being unprepared for team meetings, disrespectful or rude behavior, etc. Reasons such as “too busy” or “I forgot”, or “my dog ate my design model” are unacceptable.

Valid reasons that must be considered include those listed for obtaining an incomplete standing in a course (illness, death in the family, travel for business or academic reasons, etc.)