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

# 

# **LIST OF FIGURES**

TBD

# 

# **LIST OF TABLES**

1. Table 1.1 – Risk Analysis 4
2. Table 2.1 – Deliverables and Schedule 5
3. Table 3.1 – Evidence of Configuration Management 7
4. Table 3.2 – Evidence of Configuration Management 7

**INTRODUCTION**

This document presents the project management plan (PMP) for the development of the City Level Air Quality Prediction (CLAP) Application. Its purpose is to outline how the system will be designed, developed, and deployed, while also defining the processes and responsibilities that will guide the project. The scope of the document includes the structure, design, and management activities required to complete the CLAP system.

The CLAP system is a predictive analytics application that forecasts future Air Quality Index (AQI) categories for a single U.S. city using historical AQI and weather data from the NWS. In addition to serving as a proof of concept, CLAP can also function as an educational tool for students interested in building similar applications. By leveraging available data, the system may help anticipate air quality and related weather trends, offering insights that could support awareness and decision-making. The remainder of this document is organized as follows: Project Organization, Lifecycle Model Used, Risk Analysis, Software and Hardware Resource Requirements, Deliverables and Schedule, Monitoring, Reporting, and Controlling Mechanisms, Processional Standards, and Evidence of Configuration Management.

**PROJECT ORGANIZATION**

Team 1 (or Raytheon Group A) currently consists of six software engineers working together as a single group. This structure is intended as a temporary arrangement until the project workload can be better assessed. Since the number and complexity of modules will not be clear until detailed requirements are established, maintaining one unified team at this stage supports flexibility and adaptability. Once the architecture and design are sufficiently defined, the team will be able to divide responsibilities more effectively by forming subgroups aligned with specific modules or tasks. In the meantime, working as one group encourages communication and collaboration, ensuring that all members remain aligned during the early phases of the project.

Team Members and Roles:

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

# **LIFECYCLE MODEL USED**

The team has adopted an iterative lifecycle model to guide project development. This model emphasizes repeated cycles of design, implementation, and feedback, allowing the system to be refined and improved over time. Iteration provides opportunities for early validation of functionality and helps ensure that the system evolves in alignment with stakeholder expectations. This approach is well-suited for a proof-of-concept project, as it enables the team to demonstrate core capabilities early while progressively addressing non-functional requirements (e.g. usability, performance, and reliability).

# **RISK ANALYSIS**

# Table 1.1 – Details information regarding the likelihood and impact of risks, along with mitigation strategies.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk:** | **Likelihood:** | **Impact:** | **Mitigation:** | **Rationale:** |
| Implementation Complexity | Less than 50% | More than 50% | TBD | If the project scope expands beyond requirements, then the project will become more complex. |
| 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. |
| 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, weekly reports, communication with hosts, and clear task ownership | If the project team does not maintain communication and task ownership, then development may stall and errors increase. |

**SOFTWARE AND HARDWARE RESOURCE REQUIREMENTS**

The team intends to keep the technical environment simple and manageable to ensure successful project completion. Development, testing, and deployment will be performed using software installed on student laptops, with GitHub serving as the primary configuration management and CI/CD platform. The database will be lightweight, likely requiring one or two tables to store AQI data, and the use of cloud services may not be necessary. As a contingency measure, the team is also considering recording a demonstration video to document system functionality.

Software:

1. Python – Chosen for its accessibility and the team’s proficiency.
2. SQLite Database – Lightweight and sufficient for the project’s data storage needs.
3. GitHub – For version control, integration, and deployment management.

Hardware:

1. Student laptop – Used for development, testing, and deployment activities.

# **DELIVERABLES AND SCHEDULE**

This section outlines the project’s key deliverables and associated deadlines. Each milestone serves as an important checkpoint in the development process, ensuring that progress is made in a structured and timely manner. The timeline for future deliverables will be established once their prerequisite milestones are completed, allowing the schedule to remain flexible and aligned with project dependencies.

# Table 2.1 – Details information regarding delivery milestones, including the name of the milestone, due date, allocation of responsibility, dependencies, and estimated time.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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. | 3 hours |
| 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 |

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

Project monitoring will be achieved through a combination of weekly reports, sponsor communication, and version control practices. Weekly attendance reports will document participation in progress meetings, while weekly sponsor reports (e.g. Outlook emails) will ensure communication and alignment with project expectations. Weekly status reports will help the team meet important deadlines and maintain accountability.

* 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**

All team members are expected to uphold high professional standards throughout the project. These standards include maintaining academic integrity, demonstrating respect for one another, and ensuring an equal distribution of workload. Each member is responsible for the timely delivery of assigned tasks and is expected to contribute work of acceptable quality. Professionalism extends to behavior, such as meeting deadlines, actively participating in team activities, providing constructive feedback, and avoiding actions that could negatively impact the team’s progress.

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

The team has selected GitHub as the configuration tool for this project. The tables below provide evidence of configuration management by recording version history, authorship, and reviews of document changes. The *ID* column identifies each entry. The *date of change* column indicates when a modification was made to an existing file, and the v*ersion (before & after)* columns include the associated Git commit hash for distinction. The *author* column refers to the author of the new version. The *difference link* column provides a URL to the GitHub comparison view between two consecutive commits. The format of the difference link is as follows:

“https://github.com/cchung7/rtx\_team1/compare/<ver-before-hash>..<ver-after-hash>”.

Table 3.1 – Each entry (or row) tracks a single file revision.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID:** | **Date of Change:** | **Version Before:** | **Version After:** | **Author:** | **Review -Change Summary:** | **Reviewers:** |
| 1 | 9/10/2025 | v0.1 (3cd1923) | v0.2 (3e9a96a) | Jay Chung (cwc130330) | Made corrections to all sections. | Amelia Quinn (qcb220000) |
| 2 | 9/10/2025 | v0.2 (3e9a96a) | v0.3 (356393c) | Jay Chung (cwc130330) | Made grammatical & syntactic corrections | Amelia Quinn (qcb220000), David Santos (des210001) |
| 3 | 9/11/2025 | v0.3 (356393c) | v0.4 () | Jay Chung (cwc130330) | Made revisions to Tables. | Amelia Quinn (qcb220000), David Santos (des210001), AJ Kimbrough (ank210005) |
| 4 | TBD | TBD | TBD | TBD | TBD | TBD |

Table 3.2 – Each entry (or row) lists a difference link.

|  |  |
| --- | --- |
| **ID:** | **Difference Link:** |
| 1 | https://github.com/cchung7/rtx\_team1/compare/3cd1923..3e9a96a |
| 2 | https://github.com/cchung7/rtx\_team1/compare/3e9a96a..356393c |
| 3 | https://github.com/cchung7/rtx\_team1/compare/3e9a96a.. |
| 4 | TBD |

**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.)