

LAKEHEAD UNIVERSITY  
FACULTY OF ENGINEERING



# Software Project Management Plan

**Instructor:** Mr. Matteo Crupi

## **Group 2**

### **Full name**

---

Benita Thomas  
Fawwaz Qadeer  
Joshua Whitlock  
Kenneth Shahi

Thunder Bay, 2026

# Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
1.1	Project Summary . . . . .	2
1.1.1	Purpose, Scope, and Objectives . . . . .	2
1.1.2	Assumptions and Constraints . . . . .	2
1.1.3	Project Deliverables . . . . .	2
1.1.4	Schedule and Budget Summary . . . . .	2
1.2	Evolution of the Project Management Plan . . . . .	2
<b>2</b>	<b>Reference Materials</b>	<b>3</b>
<b>3</b>	<b>Definitions and Acronyms</b>	<b>4</b>
<b>4</b>	<b>Project Organization</b>	<b>5</b>
4.1	External Interfaces . . . . .	5
4.2	Internal Structure . . . . .	5
4.3	Roles and Responsibilities . . . . .	5
<b>5</b>	<b>Managerial Process Plans</b>	<b>6</b>
5.1	Start-up Plan . . . . .	6
5.1.1	Estimation Plan . . . . .	6
5.1.2	Staffing Plan . . . . .	6
5.1.3	Resource Acquisition Plan . . . . .	6
5.1.4	Project Staff Training Plan . . . . .	6
5.2	Work Plan . . . . .	6
5.2.1	Work Activities and Schedule Allocation . . . . .	6
5.2.2	Resource Allocation . . . . .	6
5.2.3	Budget Allocation . . . . .	7
5.3	Control Plan . . . . .	7
5.4	Risk Management Plan . . . . .	7
5.5	Project Close-out Plan . . . . .	7
<b>6</b>	<b>Technical Process Plans</b>	<b>8</b>
6.1	Process Model . . . . .	8
6.2	Methods, Tools, and Techniques . . . . .	8
6.3	Infrastructure Plan . . . . .	8
6.4	Product Acceptance Plan . . . . .	8
<b>7</b>	<b>Supporting Process Plan</b>	<b>9</b>
7.1	Configuration Management Plan . . . . .	9
7.2	Testing Plan . . . . .	9
7.3	Documentation Plan . . . . .	9
7.4	Quality Assurance Plan and Reviews and Audits Plan . . . . .	9
7.5	Problem Resolution Plan . . . . .	9
7.6	Subcontractor Management Plan . . . . .	10
7.7	Process Improvement Plan . . . . .	10
<b>8</b>	<b>Additional Plans</b>	<b>11</b>

# 1 Overview

## 1.1 Project Summary

### 1.1.1 Purpose, Scope, and Objectives

The objective of this project is to develop a recommender/explainer application that uses sensors and AI to help users grow healthy plants by providing species-specific care recommendations. The app will use the sensor data (soil moisture, air temperature, humidity, nutrient levels, pH) with plant databases and potentially AI to give actionable, context-aware advice rather than generic thresholds. Control outputs will be added, time permitting, to enable automated plant care. As this is a course based research project, the primary focus is on the research side, design artifacts and prototypes rather than a full project implementation.

### 1.1.2 Assumptions and Constraints

- The scope is limited to indoor plants or a specific region (e.g., Canadian plants) depending on data availability.
- Water quality factors (chlorine, hard water) are considered out of scope for the core system.
- AI is limited to using retrieval-augmented generation to avoid hallucinations.
- The user interface must be simple enough for non-expert gardeners.
- Class deadlines must be met.

### 1.1.3 Project Deliverables

- A prototype of a recommender/explainer app integrating sensors and AI for plant care guidance.
- Documentation (literature review, annotated bibliography, etc.).
  - Glossary of Terms.
  - Use Cases and Actors.
  - Class, Communication, Sequence UML Diagrams.
  - Specification Document.
  - Pseudocode, module outlines.
- Project Proposal (this SPMP).
- Progress Report.
- First Oral Presentation.
- Final Oral Presentation.
- Unit/Integration tests.
- Working Project Code Artifact Prototype.
- Final Report.

### 1.1.4 Schedule and Budget Summary

The duration and personnel requirements of each workflow are as follows:

- **Requirements workflow:** 2 week, four team members.
- **Analysis workflow:** 3 weeks, four team members.
- **Design workflow:** 3 weeks, four team members.
- **Implementation workflow:** 2 weeks, four team members.
- The total development time is 10 weeks.

## 1.2 Evolution of the Project Management Plan

All changes to this project management plan must be agreed upon by the entire project team and the group leader at the time before implementation. Changes should be documented to keep the plan accurate and up to date. Changes will be tracked via git commits to maintain traceability and accountability.

## **2 Reference Materials**

All artifacts will conform to the standard developed in the SPMP or in ESOF 2670. Reference materials will be constrained to articles available through the university library databases or that are freely available online with a permissive license. All external sources will be clearly cited.

### 3 Definitions and Acronyms

This list is subject to additions.

- **MVP:** Minimum Viable Product.
- **RAG:** Retrieval-Augmented Generation.
- **SPMP:** Software Project Management Plan.
- **UML:** Unified Modeling Language.

## 4 Project Organization

### 4.1 External Interfaces

All work on this project will be completed by Benita, Fawwaz, Joshua, and Kenneth. Regular communication with the course instructor will also be maintained by the appointed group leader to report progress and receive feedback.

### 4.2 Internal Structure

The development team consists of four software engineering students working together on different aspects of the project. The internal structure of the project is divided based on areas of research that are assigned during literature review. Each team member will be responsible for the artifacts of the given workflow from their area of expertise.

### 4.3 Roles and Responsibilities

Each team member has specific responsibilities to support the successful completion of the project. Because this course focuses on research, planning, and technical reporting rather than full software implementation, responsibilities are organized around project deliverables and teamwork activities. Each team member will act as project leader for a portion of the semester, coordinating meetings, delegating tasks and ensuring that deadlines are met. The role of project leader will rotate among team members as required by the course outline. This allows all members to practice leadership and planning skills. All team members share responsibility for:

- Participating actively in research and analysis.
- Contributing to written reports and presentations.
- Attending meetings and seminars.
- Reviewing each other's work.
- Meeting course and project deliverable deadlines.
- Maintaining a positive and cooperative attitude.
- Ensuring the overall quality of project deliverables.
- Documentation and code artifacts for their portion of the project for each phase of The Unified Process.

Each team member will specialize based around their area of research and progress through the phases with the team. Each team member should be interactive through the entire project, but at no point should anyone be overworked. The team will work collaboratively on all major submissions, the project lead will be responsible for distributing the workload and assembling the artifacts from each team member. Final documents will be reviewed and approved by the entire group before submission.

## 5 Managerial Process Plans

### 5.1 Start-up Plan

#### 5.1.1 Estimation Plan

The project is planned for a total duration of 10 weeks, aligned with the Unified Process workflows already defined for this course project. The estimate is based on expert judgment and analogy with previous student software projects of similar scope, and it is expressed primarily in time (person-hours) rather than money because this is a course project. The work is divided into requirements, analysis, design, and implementation. Each workflow has clear deliverables (use cases, architecture/design artifacts, and a tested prototype) that group members must complete.

#### 5.1.2 Staffing Plan

The project will be completed by a four-member team. The leadership rotates as required by the course outline. The appointed group leader will be responsible for coordinating meetings, keeping meeting minutes, delegating tasks, and ensuring that deadlines are met. To keep responsibilities clear and prevent overlap, each member will take point on their area of expertise, decided during the research phase. Each team member will be responsible for the artifacts developed during each phase, and the group leader will be responsible for integrating all artifacts into a complete, cohesive version for group review. All members participate in testing and review of the artifacts after they have been compiled by the group leader.

#### 5.1.3 Resource Acquisition Plan

The project requires basic hardware for data collection and standard software development tools. Hardware includes a Raspberry Pi (or equivalent), a soil moisture sensor (preferably capacitive), a temperature/humidity sensor, and a light sensor. Optional sensors such as EC (nutrient proxy) and pH may be added if time allows, but they are not required for the minimum viable system. Additional items include a breadboard, resistors (if needed), and a stable power supply.

Each team member has access to a desktop or laptop computer on which to write code and update documentation using their chosen IDE. Software resources include Git and a Git hosting platform (GitHub) for version control, a code editor (e.g., VS Code), and a lightweight database. The project will be implemented in Python, and the project will maintain clear setup instructions so the system can be run on another machine.

#### 5.1.4 Project Staff Training Plan

No formal external training is required. The team will review how to structure plant profiles and how to convert raw readings into actionable recommendations. Any unfamiliar tools (e.g., Markdown documentation, Git workflow, or simple API development) will be learned through development of small prototypes.

### 5.2 Work Plan

#### 5.2.1 Work Activities and Schedule Allocation

The work is organized into five workflows across 10 weeks:

- **Week 1-2 (Requirements):** write and hand in the complete SPMP, build a Glossary of Terms, define Use Cases and Actors.
- **Weeks 3-5 (Analysis):** refine use cases, build initial class, communication, sequence diagrams in UML format. Write the Specification Document.
- **Weeks 6-8 (Design):** complete class diagrams, finalize attributes and methods, update class UML diagrams. Outline modules and write pseudocode.
- **Weeks 9-10 (Implementation):** build unit and integration tests, write the first prototype code.

#### 5.2.2 Resource Allocation

Team members will work in parallel but integrate continuously. Each team member will be responsible for the artifacts from his/her area of research, and the team lead is responsible for ensuring the artifacts are integrated properly. Furthermore, each team member is required to signoff on each artifact after the group leader has integrated them. Weekly coordination will include a short team meeting to confirm progress, update tasks, and identify blockers. Task tracking will be done using Git issues (or an equivalent board).

### 5.2.3 Budget Allocation

This is a student project, the budget is mainly limited to low-cost sensors and basic electronics accessories. The plan is to reuse any available lab equipment and purchase only essential components. The minimum hardware set is kept intentionally small (Raspberry Pi + moisture + temperature/humidity + light) so the project remains feasible even if optional sensors (EC/pH) are not acquired or are unreliable. Any costs will be tracked and kept minimal, and the software stack will use open-source tools.

### 5.3 Control Plan

Project control will be handled through artifact-based tracking and strict version control practices. Each workflow has defined outputs. Changes to scope must be agreed upon by the full team and only accepted by the team leader if they do not threaten the schedule. All code and documentation changes will be made through Git commits, and major changes will be made on feature branches and merged via pull requests with at least one reviewer. The integration lead will ensure that the main branch remains buildable and that unfinished work is isolated to branches. Weekly checkpoints will confirm that progress matches the schedule and that the project remains aligned with the stated objectives.

### 5.4 Risk Management Plan

Key risks and mitigation strategies are as follows:

- **Sensor inaccuracy or instability:** mitigate by using a capacitive moisture sensor, validating readings over time, and documenting calibration assumptions. Implement sanity checks (out-of-range detection) to prevent bad recommendations.
- **Data availability for species-specific guidance:** mitigate by using a plant profile dataset that can be downloaded or curated, and limiting the first release to a defined set of plant species with reliable references.
- **Scope creep (too many sensors or features):** mitigate by defining an MVP (moisture + temperature/humidity + light + species profiles) and treating extra sensors/features as stretch goals.
- **Integration delays:** mitigate by integrating early and using mocked data so the UI and backend can progress while hardware is being stabilized.
- **Recommendation errors causing misleading advice:** mitigate by making recommendations explainable (show the measured value, the target range for the species, and the action) and adding conservative messaging when confidence is low.
- **Hardware procurement delays or failures:** mitigate by ordering early and keeping a software-only simulation mode to continue development without hardware.

### 5.5 Project Close-out Plan

Project close-out will include a final integrated build, a demonstration of sensor-to-recommendation functionality, and a complete repository containing the code, setup instructions, and documentation. The team will tag a final release in the repository and ensure that the README provides step-by-step instructions to run the system. A brief post-project review will capture lessons learned (what worked well, what was difficult, and what could be improved in a future iteration).



## 6 Technical Process Plans

### 6.1 Process Model

The Unified Process will be used.

### 6.2 Methods, Tools, and Techniques

- **Workflows:** All workflows will follow the Unified Process.
- **Hardware:** Raspberry Pi will be used for sensor integration; sensors may include soil moisture, air temperature, humidity, nutrient/EC and pH sensors.

### 6.3 Infrastructure Plan

- **Software/Languages:** Python or similar for app development; AI integration using retrieval-augmented generation with plant databases to avoid hallucinations.
- **APIs/Databases:** External plant database APIs for species-specific information; manual user input for species identification if needed.

### 6.4 Product Acceptance Plan

Acceptance of the product will be achieved by following the steps of the Unified Process, including phase deliverables such as use cases, class diagrams, working prototype, and testing.

## 7 Supporting Process Plan

### 7.1 Configuration Management Plan

GitHub will be used as the main configuration management tool.

The following practices will be followed:

- All source code and documents will be stored in a shared GitHub repository.
- Team members will work on separate branches for new features.
- Pull requests will be used before merging changes.
- The main branch will always remain stable.
- Regular commits will be required from all members.

This approach ensures that all project artifacts are organized, traceable, and properly backed up.

### 7.2 Testing Plan

Testing will be performed throughout the project to ensure reliability.

The testing process will include:

- **Reviews:** artifacts generated during the Requirements, Analysis and Design phases will undergo the review process.
- **Unit Testing:** individual software components will be tested separately.
- **Integration Testing:** sensor hardware and software modules will be tested together.
- **System Testing:** the full system will be tested from data collection to recommendations.
- **User Testing:** team members will evaluate usability and clarity of outputs.

Test results will be documented and stored in the project repository.

### 7.3 Documentation Plan

Clear documentation will be produced at every stage of the project.

The main documents include:

- Meeting minutes recorded weekly on D2L.
- The Software Project Management Plan (this document).
- Design and Architecture UML documents.
- User Guide and Setup Instructions.
- Test reports.
- A progress report.
- Slides for *both* oral presentations.
- The final Project Report.

All documentation will be reviewed and updated regularly to ensure accuracy and consistency.

### 7.4 Quality Assurance Plan and Reviews and Audits Plan

Quality will be ensured through:

- Regular peer reviews of code and documents.
- Weekly team meetings to track progress.
- Continuous testing and debugging.
- Instructor feedback and reviews.

No major component will be considered complete without review by at least one other team member.

### 7.5 Problem Resolution Plan

If a problem occurs during the project:

- The issue will be discussed in team meetings.
- The responsible member will attempt to fix it.

- If needed, work will be reassigned.
- Serious problems will be reported to the instructor, Mr. Matteo.

All important problems will be tracked using GitHub issues so they can be resolved in an organized way.

## **7.6 Subcontractor Management Plan**

This project does not involve any subcontractors. All work will be completed by the student project team.

## **7.7 Process Improvement Plan**

The team will continuously improve its working process by:

- Reviewing progress after each milestone.
- Adjusting schedules when necessary.
- Improving communication methods.
- Learning from mistakes and feedback.

At the end of the project, a final review will be conducted to capture lessons learned. Minutes will be tracked for this and a document will be generated for each group member to take something away. At the end of each phase, the group member should also conduct a postmortem of what went right and what could've gone better.

## 8 Additional Plans

- **Security:** the system will avoid storing sensitive personal data. If user accounts are implemented, passwords will be stored securely (hashed), and API keys/secrets will not be committed to the repository (group members will use proper `.gitignore` management practices).
- **Data/licensing:** plant databases and any external datasets used will be cited, and licensing constraints will be respected. If a dataset cannot be legally reused, the team will replace it with an approved alternative or a curated set with proper references.