Project Management Plan

This Project Management Plan outlines the approach and methodologies to be used in the development of GrowHub MVP. The project aims to deliver a functional prototype with integrated sensors, peripheral devices, and a web-based dashboard within an 8-week timeframe. Glossary of used terms, acronyms, and abbreviations is included at the end of the document.

Version: 001
Created: 7. 7. 2024 by Filip Vallo (Project Manager)
Approved: 7. 7. 2024

1. GENERAL PROJECT INFORMATION

Project name:
GrowHub

Project Manager: Filip Vallo
Email: filip.vallo@gmail.com

Expected start date: 8. 7. 2024
Expected completion date: 31. 8. 2024

2. PROJECT OVERVIEW

2.1 Description

GrowHub is an MVP IoT product designed to help indoor growers monitor and control their growing environments. The system will collect real-time data from various sensors and provide users with a web-based dashboard for monitoring and managing growing conditions. The product will include a set of fully programmable peripheral devices for irrigation control, fertilizer injection, and power control. The dashboard will also allow to set alerts with notifications if any of the measured parameters exceeds set thresholds.

2.2 Purpose and Justification

The purpose of the GrowHub project is to develop an MVP IoT solution that provides hobbyists and professional growers with the tools to monitor and control their indoor growing environments. This solution aims to enhance plant growth by maintaining optimal conditions through automated and precise control of environmental factors.

2.3 Problem Statement

Indoor growers face challenges in maintaining optimal growing conditions for their plants. The lack of precise and automated control over environmental factors can lead to suboptimal plant growth and reduced yields.

2.4 Business Case

Developing the GrowHub MVP will demonstrate the feasibility of an IoT-based solution for indoor growing environments. This project has the potential to attract commercial interest, leading to further development and potential market entry. Successful implementation of the MVP will showcase the value of precise environmental control in improving plant health and yields.

2.5 Vision

Revolution of indoor growing through an innovative IoT solution that empowers both hobbyists and professionals to achieve optimal plant health and yields, by providing precise, automated environmental control and real-time monitoring.

2.6 Goals

- Manufacture a functional prototype of the monitoring IoT system using Raspberry Pi with integrated sensors to measure atmospheric and soil conditions
- Manufacture connected programmable peripheral IoT devices for irrigation control, fertilizer injection, and power control
- Develop a fully operational website with dashboard for live and historical data visualization with notification system and environmental control system for managing growing conditions
- Manufacture embedded PCB that integrates all electronic circuits needed to process digital signals from all connected sensors and to control all custom IoT peripheral devices
- Test the MVP's commercial potential

2.7 Scope Statement

The GrowHub project encompasses the development, testing, and deployment of a fully functional MVP IoT system for indoor growing environment monitoring and management.

2.8 Functional Requirements

2.8.1 IoT device

- Real-time data collection from atmospheric sensors (temperature, humidity, CO2) and soil sensors (moisture, pH, NPK) connected to Raspberry Pi microcomputer
- Fully programable peripheral IoT devices for irrigation control, fertilizer injection, and power control
- Embedded PCB with custom integrated electronic circuit that processes digital signals from all connected sensors and controls all peripheral devices

2.8.2 Web-based dashboard

- Real-time data collection from IoT device
- Historical data tracking and visualization
- Notification system for alerts when measured parameters exceed thresholds
- Environmental control system for managing growing conditions through peripheral devices

2.9 Non-functional Requirements

- High reliability and accuracy of sensor data
- Secure and responsive web dashboard
- User-friendly interface for both novice and experienced growers
- Scalability to handle multiple users and devices in the future

2.10 Outside of Scope

- Full commercialization and mass production (beyond MVP)
- User registration/authentication and user management system
- Advanced features not essential for the MVP
- Casing and product design

2.11 Risks

- Technical challenges in integrating sensors and developing the dashboard
- Limited budget and resources
- Delays in hardware procurement
- Delays in custom PCB manufacturing from the external vendor
- Quality issues with the manufactured PCB

2.12 Constraints

- Limited budget for hardware procurement and development
- Tight timeline to complete the MVP within the four planned phases
- Dependence on third-party services for cloud hosting and data storage
- Dependence on external vendor timelines for custom PCB manufacturing

2.13 Assumptions

- Availability of necessary hardware components
- Access to development tools and platforms for building the website
- The external vendor will meet the specified quality and timeline requirements for the custom PCB circuit
- Initial funding is sufficient to develop the MVP

3. PROJECT ORGANIZATION

3.1 Project Team Structure

As GrowHub is a personal project, Filip Vallo will be assuming all roles:

- Project Sponsor, Project Manager, IoT HW Engineer, SW Architect, SW Developer

3.2 Roles and Responsibilities

Filip Vallo will be responsible for all aspects of the project, including:

- Project management and decision-making
- Hardware design and integration
- Software architecture and development
- Testing and quality assurance
- Stakeholder communication and management

4. PROJECT MANAGEMENT APPROACH

4.1 Project Lifecycle

The project will follow an Agile approach, utilizing 1-week Sprints within four main phases corresponding to the key deliverables. This approach allows for iterative development, frequent reassessment of priorities, and regular delivery of incremental value, enabling continuous improvement and adaptation throughout the project lifecycle.

4.2 Project Structure

- The project is divided into four main phases, each focusing on a key component of the GrowHub system
- Within each phase, work will be organized into 1-week Sprints
- Each Sprint will follow the Scrum framework, including Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective

4.3 Management Processes

- **Planning**: Weekly Sprint Planning and ongoing project backlog refinement
- **Execution**: Daily development activities guided by Sprint goals
- **Monitoring/Controlling**: Daily Scrums, weekly Sprint Reviews and Retrospectives
- **Closing**: Final testing, documentation, and project wrap-up during the last Sprint

5. SCOPE MANAGEMENT PLAN

5.1 Scope Definition

The scope of the GrowHub project is defined in the Project Charter and includes the development of an MVP IoT product for monitoring and controlling indoor growing environments.

5.2 Work Breakdown Structure (WBS)

A high-level WBS will be created and maintained in a separate document. The WBS will break down the project into manageable work packages aligned with project structure (four main phases, two Sprints for each phase).

5.3 Scope Baseline

The scope baseline consists of the approved Project Charter, WBS, and WBS Dictionary. These documents will serve as the foundation for scope management and control.

5.4 Scope Control

Scope will be controlled through the following mechanisms:

- **Product Backlog**: Maintained and prioritized regularly to reflect current project needs and constraints
- **Sprint Planning**: Scope for each Sprint will be defined and agreed upon at the beginning of each week
- **Sprint Review**: At the end of each sprint, completed work will be reviewed against the Sprint goals and Definition of Done
- **Definition of Done**: Clear criteria will be established for when a feature or component is considered complete

5.5 Scope Changes

Given the MVP nature of the project and the Agile approach, some scope flexibility is expected. Scope changes will be managed as follows:

- **Minor Changes:** Small adjustments that don't affect the overall project objectives can be incorporated into the Product Backlog and prioritized accordingly
- **Major Changes**: Significant changes that impact project timeline or budget will require a formal change request and approval process from wife

5.6 Scope Verification

Scope verification will occur at multiple levels:

- **Daily**: Through Daily Scrum meetings to ensure work is progressing as planned
- **Weekly**: During Sprint Reviews to verify that completed work meets the Definition of Done and project requirements
- **Phase Completion**: At the end of each major phase to ensure all components are integrated and functioning as expected

5.7 Scope Management Roles and Responsibilities

As GrowHub is a personal project, Filip Vallo will be responsible for all aspects of scope management, including:

- Maintaining the Product Backlog
- Defining and prioritizing Sprint goals
- Conducting Sprint Plannings, Sprint Reviews, and Sprint Retrospectives
- Managing and approving scope changes
- Verifying completed work against project requirements

5.8 Scope Management Tools and Techniques

The following tools and techniques will be used for scope management:

- **Product Backlog**: Maintained in a digital task management tool
- **Epics**: Used to define large body of work
- **User Stories**: Used to define features and functional requirements
- **Tasks**: Used to define specific pieces of technical work
- **Sprint Burndown Charts**: To track progress within each sprint
- **Definition of Done**: To ensure quality and completeness of deliverables

6. SCHEDULE MANAGEMENT PLAN

6.1 Schedule Methodology

The GrowHub project will use an Agile scheduling approach, aligning with the overall project management methodology. The schedule will be based on 1-week Sprints within four main project phases.

6.2 Schedule Tools

A digital project management tool will be used to manage the product backlog, Sprint backlogs, and high-level project timeline.

6.3 Schedule Development

The project schedule will be developed and maintained at two levels:

6.3.1 High-level Project Timeline

- Outlines the four main project phases
- Identifies key milestones and deliverables
- Spans the entire project duration (July 8, 2024 - August 31, 2024)

6.3.2 Sprint-level Schedule

- Detailed schedule for each 1-week Sprint
- Developed during Sprint Planning at the beginning of each week

6.4 Project Phases and Timelines

The project is divided into four main phases, each consisting of two 1-week Sprints and expected deliverable at the end of each phase:

- **Phase 1**: 08. 07. 2024 – 21. 07. 2024
- **Phase 2**: 22. 07. 2024 – 04. 08. 2024
- **Phase 3**: 05. 08. 2024 – 18. 08. 2024
- **Phase 4**: 19. 08. 2024 – 01. 09. 2024

6.5 Expected Deliverables

- **Phase 1**: Prototype of monitoring IoT device with sensors
- **Phase 2**: Web-based dashboard with data collection from sensors, historical data tracking, visualization, and notification system
- **Phase 3**: Prototypes of peripheral devices for irrigation control, fertilizer injection, and power control; integrated embedded PCB
- **Phase 4**: Web-based dashboard with environmental control system for managing growing conditions through peripheral devices

6.6 Regular Events

Each 1-week Sprint will follow this structure:

- **Sprint Planning**: Beginning of each week
- **Daily Scrum**: Brief daily check-ins
- **Sprint Review** and **Sprint Retrospective**: End of each week

6.7 Schedule Control

Schedule control will be maintained through the following mechanisms:

- **Daily Scrums**: Brief daily check-ins to discuss progress and identify any scheduling issues
- **Sprint Reviews**: Weekly reviews to assess completed work against Sprint goals
- **Sprint Retrospectives**: Weekly retrospectives to identify process improvements, including scheduling efficiencies
- **Sprint Burndown Charts**: Used to track progress within each Sprint

6.8 Schedule Changes

Given the Agile nature of the project, the schedule is expected to be flexible within the overall project timeline. Schedule changes will be managed as follows:

- **Sprint-level Changes**: Adjustments to tasks within a Sprint can be made during Daily Scrums, as long as Sprint goals are not compromised
- **Phase-level Changes**: Changes that affect the completion of a phase or shift work between phases will be assessed for their impact on the overall project timeline and documented
- **Project-level Changes**: Major changes that affect the overall project timeline will require a formal change request and approval process from wife

6.9 Schedule Performance Measurement

Schedule performance will be measured using the following metrics:

- **Sprint Completion Rate**: Percentage of planned sprint tasks completed
- **Velocity**: Amount of work completed in each Sprint, measured in story points or number of tasks
- **Milestone Completion**: Timely achievement of key project milestones

6.10 Schedule Reporting

Schedule status and performance will be self-reported and documented at the following intervals:

- **Daily**: Brief notes on progress and any identified scheduling issues
- **Weekly**: Sprint review report summarizing completed work and any schedule variances
- **Phase Completion**: Report on phase outcomes, including schedule performance and any lessons learned

6.11 Schedule Management Roles and Responsibilities

As GrowHub is a personal project, Filip Vallo will be responsible for all aspects of scheduling management, including:

- Developing and maintaining the high-level project timeline
- Conducting Sprint Planning and defining Sprint backlogs
- Tracking progress and updating Sprint burndown charts
- Identifying and managing schedule risks and changes
- Conducting Sprint Reviews and Sprint Retrospectives

7. COST MANAGEMENT PLAN

7.1 Cost Management Approach

The GrowHub project will use a fixed-budget approach, with costs managed within the initial budget of €1500. Cost management will focus on optimal allocation of this budget across project needs, tracking expenses, and ensuring the project remains within budget constraints.

7.2 Cost Estimation

Initial cost estimates have been provided in the Project Charter. These estimates will be refined as the project progresses, particularly at the beginning of each project phase.

7.3 Budget Allocation

The initial budget of 1500 EUR is allocated as follows:

- **Development tools**: 200 EUR
- **Cloud services**: 200 EUR
- **Hardware components**: 500 EUR
- **Supplies**: 200 EUR
- **Outsourcing** (PCB manufacturing): 200 EUR
- **Miscellaneous**: 200 EUR

7.4 Cost Control

Cost control will be maintained through the following mechanisms:

- **Expense Tracking**: All expenses will be recorded on a spreadsheet, categorized according to the budget allocation
- **Regular Review**: Costs will be reviewed at the end of each Sprint to ensure alignment with the budget
- **Forecasting**: Remaining costs will be forecasted at the end of each project phase to identify any potential budget overruns early

7.5 Cost Changes and Thresholds

Given the fixed budget, cost changes will be managed as follows:

- **Reallocation**: Funds can be reallocated between categories if needed, as long as the total budget remains unchanged
- **Overspend Threshold**: Any potential overspends exceeding 15% in any category must be identified and addressed immediately
- **Budget Increase**: Any need for budget increase must go through a formal change request process and approval process from wife, including justification and impact assessment

7.6 Cost Performance Measurement

Cost performance will be measured using the following metrics:

- **Actual Cost** (**AC**): The actual cost incurred for work performed
- **Planned Value** (**PV**): The budgeted cost of work scheduled
- **Earned Value** (**EV**): The budgeted cost of work performed
- **Cost Performance Index** (**CPI**): Calculated as EV/AC, indicating cost efficiency

7.7 Cost Reporting

Cost status and performance will be self-reported and documented at the following intervals:

- **Weekly**: Brief cost summary as part of the Sprint review, including any significant expenses or variances
- **Phase Completion**: Detailed cost report including actual costs, variances, and forecasts for the remaining project
- **Project Completion**: Final cost report comparing actual costs to the initial budget, analyzing variances, and documenting lessons learned

7.8 Cost Management Roles and Responsibilities

As GrowHub is a personal project, Filip Vallo will be responsible for all aspects of cost management, including:

- Tracking all project expenses
- Reviewing and approving all purchases
- Monitoring cost performance
- Identifying and managing cost risks
- Preparing cost reports

7.9 Cost Management Tools and Techniques

The following tools and techniques will be used for cost management:

- **Spreadsheet**: To track expenses and calculate cost performance metrics
- **Earned Value Management** (**EVM**): To integrate scope, schedule, and cost performance
- **Three-Point Estimating**: For more accurate cost estimations when needed

7.10 Procurement Management

For procured items (e.g., hardware components, PCB manufacturing):

- At least three quotes will be obtained for any item exceeding 100 EUR in cost
- All purchases will be documented with receipts and entered into the expense tracking spreadsheet

8. QUALITY MANAGEMENT PLAN

8.1 Quality Management Approach

The GrowHub project will adopt a quality-focused approach throughout its development cycle, emphasizing continuous improvement and iterative testing. The goal is to ensure that the MVP meets functional requirements and provides a reliable, user-friendly experience for indoor growers.

8.2 Quality Objectives

- Ensure accurate and reliable data collection from sensors
- Develop a responsive and intuitive web dashboard
- Create robust and reliable peripheral devices for environmental control
- Integrate all components into a cohesive, functional system
- Maintain code quality and documentation standards
- Ensure user safety in the operation of the device

8.3 Quality Standards

8.3.1 Hardware Standards

- Sensor accuracy within ±5% of actual values
- Peripheral devices respond to commands within 2 seconds
- PCB design adheres to IPC-2221 standards

8.3.2 Software Standards

- Web dashboard loads within 3 seconds on standard broadband connection
- Code adheres to PEP 8 style guide for Python
- Front-end follows WCAG 2.1 Level AA accessibility guidelines

8.3.3 System Integration Standards

- Data synchronization between IoT device and dashboard within 5 seconds

8.3.4 Documentation Standards

- Code is commented with docstrings for all functions and classes
- README files are provided for each major component
- User manual includes clear setup and operation instructions

8.4 Quality Control Measures

- **Definition of Done** (**DoD**): Clear criteria for when a feature is considered complete
- **Checklists**: Checklists for common processes
- **Test-Driven Development** (**TDD**): For critical software components
- **Continuous Integration**: Automating build and basic test processes where possible

8.5 Quality Monitoring and Control

- **Sprint Reviews**: Assessing completed work against quality standards
- **Defect Tracking**: Logging and prioritizing of any identified issues or defects
- **Metrics Tracking**: Defect density (number of defects per unit of work), Test pass rate and Code coverage for automated tests

8.6 Quality Improvement Process

- **Sprint Retrospectives**: Event to identify areas for quality improvement
- **Root Cause Analysis**: Techniques to identify the fundamental reasons behind identified defects or issues (for any significant quality issues)
- **Iterative Refinement**: Continuous update of quality standards and processes based on learnings

8.7 Quality Roles and Responsibilities

As GrowHub is a personal project, Filip Vallo will be responsible for all aspects of quality management, including:

- Defining and updating quality standards
- Conducting all testing and quality assurance activities
- Tracking quality metrics and defects
- Implementing quality improvements

8.8 Quality Tools and Techniques

- **Version Control**: Source code management and tracking of changes in documentation
- **Code Reviews**: Systematic examination of code changes
- **Static Code Analysis**: Automated code review techniques to identify potential issues in the source code
- **Manual Testing**: Comprehensive testing of user interfaces and hardware interactions
- **Automated Testing**: Automated tests to verify the functionality of critical components (unit testing, integration testing, and system tests)
- **User Testing**: Simulated use-case testing at the end of each major phase
- **Continuous Integration**: Practice of frequently merging code changes into a central repository, after which automated builds and tests are run
- **Defect Tracking**: System to log, prioritize, and track the resolution of issues or bugs
- **Documentation Review**: Regular reviews and updates of project documentation

8.9 Quality Documentation

- **Test Plans**: Outlining of test cases for each major component
- **Test Reports**: Results of manual and automated testing
- **Defect Logs**: Tracking of identified issues and their resolutions
- **Quality Metrics Dashboard**: Visual representation of key quality metrics

9. RESOURCE MANAGEMENT PLAN

9.1 Resource Management Approach

The GrowHub project will primarily rely on the skills and time of a single team member, Filip Vallo. The focus will be on efficient time management, strategic skill development, and optimal use of physical and virtual resources.

9.2 Human Resources

9.2.1 Roles and Responsibilities

Filip Vallo will fulfill all project roles, including:

- Project Manager
- IoT Hardware Engineer
- Software Architect
- Software Developer
- Quality Assurance Engineer

9.2.2 Time Management

- **Work Schedule**: 40 hours per week dedicated to the project
- **Time Tracking**: Daily logging of hours spent on different project activities

9.2.3 Time Allocation

- **Development** (Hardware and Software): 75%
- **Project Management and Planning**: 10%
- **Documentation and Reporting**: 10%
- **Testing and Quality Assurance**: 5%

9.2.4 Skill Development

Areas where skill enhancement is needed:

- **PCB Design**: Allocating time for research
- **Web Development**: Focusing on improving React and dashboard design skills
- **IoT Protocols**: Studying MQTT and other relevant protocols

9.3 Physical Resources

9.3.1 Hardware Components

- Raspberry Pi microcomputer
- Various sensors (temperature, humidity, CO2, moisture, pH, NPK)

9.3.2 Supplies

- Wiring
- Breadboards
- Soldering material
- Various microelectronic components and connectors

9.3.3 Tools and Machinery

- Workstation
- Soldering station
- Testing equipment

9.3.4 Facilities

- Controlled environment to test the sensors and peripheral devices

9.3.5 Resource Acquisition

- Detailed listing of required components
- Ordering of components in advance to prevent delays
- Considering local electronics stores for urgent needs

9.4 Virtual Resources

9.4.1 Software

- Software development tools
- Management and issue tracking tools
- Cloud services

9.4.2 Licenses and Subscriptions

- Maintaining a list of necessary software licenses
- Tracking subscription renewals for cloud services

9.5 External Resources

9.5.1 PCB Manufacturing

- Researching and selecting a reliable PCB manufacturer
- Planning for manufacturing lead times in the project schedule

9.5.2 Cloud Services

- Selecting and setting up appropriate cloud services
- Monitoring usage to stay within allocated budget

9.6 Resource Allocation

- Aligning resource allocation with the project schedule and Sprint plans
- Prioritizing critical path activities in resource allocation

9.7 Resource Management Tools and Techniques

- **Time Tracking**: Logging of daily activities
- **Kanban Board**: Visualizing workflow and resource allocation
- **Resource Calendar**: Mapping out availability of physical resources and external services

9.8 Resource Risk Management

- Identifying potential resource constraints or bottlenecks
- Developing contingency plans for critical resources
- Monitoring for signs of burnout and implement stress-management techniques

9.9 Resource Performance Measurement

- Tracking actual time spent versus estimated time for activities
- Monitoring progress against skill development goals
- Assessing efficiency of physical and virtual resource usage

9.10 Resource Management Review

- Conducting a weekly review of resource utilization
- Adjusting allocation and management strategies as needed
- Documenting lessons learned for future projects

10. STAKEHOLDER ENGAGEMENT PLAN

10.1 Stakeholder Engagement Approach

The GrowHub project will adopt a proactive stakeholder engagement approach, focusing on clear communication, expectation management, and value creation. While primarily a solo project, this plan will address self-management strategies and prepare for potential future stakeholder interactions. All stakeholders and their relevant information will be recorded and kept updated in Stakeholder Register.

10.2 Stakeholder Identification and Analysis

10.2.1 Project Team

- **Filip Vallo** (Project Manager, IoT HW Engineer, SW Architect, SW Developer)
  - Interest: Project success, skill development, potential future commercialization
  - Influence: High
  - Engagement Level: Leading

10.2.2 Current Stakeholders

10.2.3 Future Stakeholders

- **External PCB Manufacturer**
  - Interest: Clear specifications, timely payment
  - Influence: Low
  - Engagement Level: Neutral (keep informed)
- **Cloud Service Provider**
  - Interest: Service utilization, timely payment
  - Influence: Low
  - Engagement Level: Neutral (monitor)

10.2.4 Potential Future Stakeholders

- **Potential investors**
  - Interest: Project viability, total addressable market (TAM), return on investment (ROI)
  - Influence: High
  - Engagement Level: Unaware (keep satisfied)
- **Potential commercial partners**
  - Interest: Product functionality, market fit, scalability
  - Influence: Medium
  - Engagement Level: Unaware (keep informed)

10.2.5 Target users

- **Hobbyist indoor growers**
  - Interest: Product functionality, user experience, value proposition
  - Influence: High
  - Engagement Level: Unaware (keep satisfied)
- **Professional indoor growers**
  - Interest: Product functionality, user experience, value proposition
  - Influence: High
  - Engagement Level: Unaware (keep satisfied)

Stakeholder	Role	Project's Impact	Influence on Project
Filip Vallo	Project Sponsor, Project Manager, IoT HW Engineer, SW Architect, SW Developer	High	High
External PCB Manufacturer	Vendor for custom PCB	Low	Low
Cloud Service Provider	Infrastructure for web dashboard	Low	Low
Potential Investors	Possible future funding	High	High
Potential Commercial Partners	Possible collaboration or distribution	High	Medium
Hobbyist Indoor Growers	Target users	High	High
Professional Indoor Growers	Target users	Medium	High

10.3 Communication Objectives

- Maintaining clear and organized project documentation
- Tracking project progress effectively
- Preparing for potential future stakeholder communications
- Ensuring clear communication of project outcomes and lessons learned

10.4 Communication Methods

- **Project Documentation**: Compiling, recording, and sharing documentation
- **Code Documentation**: Inline comments and README files
- **Progress Tracking**: System to log, prioritize, and track the resolution of requirements
- **Personal Project Journal**: Digital document or notebook for daily reflections
- **External Communications**: Email client or phone

10.5 Guidelines for Communication Activities

10.5.1 Daily Scrum (Self-review)

- Review of tasks completed day before
- Planning of tasks for today
- Identifying any blockers or issues
- Updating project management tool

10.5.2 Sprint Review and Retrospective

- Review of completed work against Sprint goals
- Document progress, challenges, and lessons learned
- Planning of improvements for the next Sprint

10.5.3 Phase Completion Report

- Summary of phase achievements
- Documenting any deviations from the plan
- Updating project risks and mitigation strategies
- Revision of project plan if necessary

10.5.4 External Communications

- Maintaining a professional tone in all external communications
- Keeping a record of all communications with external parties
- Preparing templates for common communications (e.g., PCB order emails)

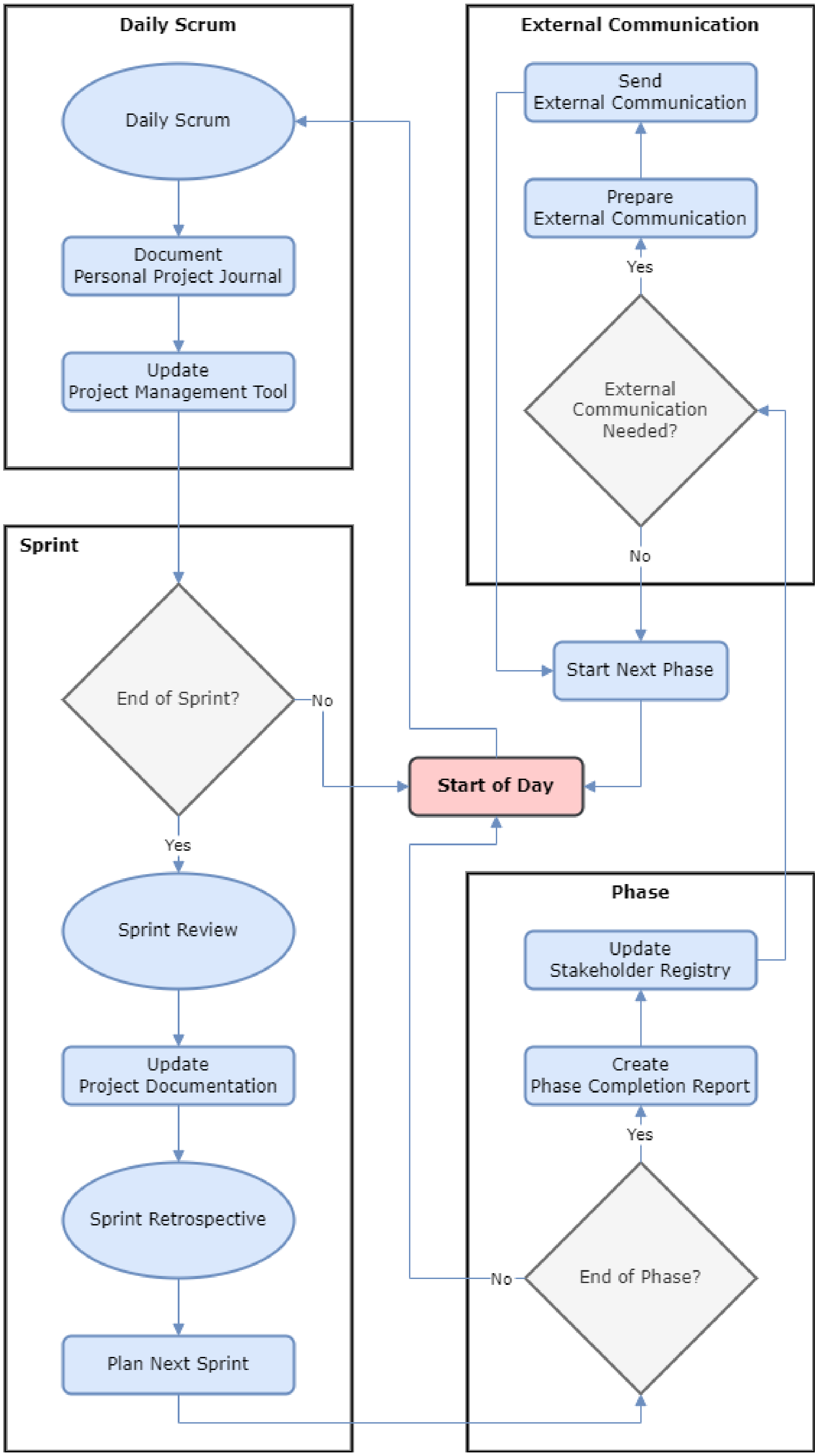
10.6 Communication Standards

- Using clear and concise language in all documentation
- Maintaining consistent formatting in project documents
- Using version control for all project artifacts
- Including date and version number on all formal documents

10.7 Stakeholder Communication Plan

Information	Stakeholders	Method	Frequency	Owner
Project status	Project Team	Project management tool	Daily	Filip Vallo
Sprint Review	Project Team	Self-conducted review and documentation	Weekly	Filip Vallo
Phase completion	Project Team	Formal document	End of each phase	Filip Vallo
Project outcome and financial projections	Potential Investors, Potential Commercial Partners	Presentation / demo	Project completion	Filip Vallo
User manual and setup guide	Hobbyist Indoor Growers, Professional Indoor Growers	User documentation	Project completion	Filip Vallo
Design for custom PCB	External PCB Manufacturer	Email	As needed	Filip Vallo
Service requirements for web dashboard	Cloud Service Provider	Cloud service	As needed	Filip Vallo

10.8 Stakeholder Communication Flowchart


Daily Scrum
Daily Scrum
Document Personal Project Journal
Update Project Management Tool
External Communication
Send External Communication
Prepare External Communication
Yes
External Communication Needed?
No
Start Next Phase
Sprint
End of Sprint?
No
Yes
Start of Day
Sprint Review
Update Project Documentation
Sprint Retrospective
Plan Next Sprint
Phase
Update Stakeholder Registry
Create Phase Completion Report
Yes
No
End of Phase?

10.9 Stakeholder Engagement Strategies

10.9.1 Project Team

- **Filip Vallo** (Project Manager, IoT HW Engineer, SW Architect, SW Developer)
  - Maintaining a project journal for daily reflections and progress tracking
  - Conducting weekly self-reviews aligned with Sprint reviews
  - Setting clear personal milestones and celebrate achievements
  - Engaging with online developer communities for support and knowledge sharing

10.9.2 Current Stakeholders

10.9.3 Future Stakeholders

- **External PCB Manufacturer**
  - Providing clear, detailed specifications for PCB manufacturing
  - Maintaining professional communication and adhering to agreed timelines
  - Seeking feedback on design for manufacturability
- **Cloud Service Provider**
  - Regular reviews of service utilization and costs
  - Staying informed about service updates or changes

10.9.4 Potential Future Stakeholders

- **Potential investors**
  - Preparing a compelling project presentation and demo
  - Developing a clear business case and market analysis
  - Maintaining detailed project documentation and financial projections
- **Potential commercial partners**
  - Preparing a compelling project presentation and demo
  - Creating a Product Specifications document to highlight key features and benefits
  - Preparing use case scenarios demonstrating product versatility
  - Developing a scalability plan for potential mass production

10.9.5 Target users

- **Hobbyist indoor growers**
  - Developing user personas to guide product development
  - Creating user documentation and setup guides
  - Planning for potential beta testing or user feedback sessions on MVP features and usability
- **Professional indoor growers**
  - Developing user personas to guide product development
  - Creating user documentation and setup guides
  - Planning for potential beta testing or user feedback sessions on MVP features and usability
  - Seeking input on advanced features and scalability

10.10 Monitoring Stakeholder Engagement

- Regular review and update of the Stakeholder Register
- Assessing the effectiveness of engagement strategies through self-reflection and any external feedback
- Adjusting engagement approaches based on project progress and stakeholder dynamics

10.11 Stakeholder Engagement Success Metrics

- Self-assessment of project satisfaction and motivation levels
- Timely and successful completion of PCB manufacturing
- Quality of prepared materials for potential future stakeholders
- Effectiveness of user documentation (to be assessed in potential future user testing)

10.12 Continuous Improvement

- Regular reflections on stakeholder engagement strategies and their effectiveness
- Seeking opportunities for improving communication and engagement methods
- Documenting lessons learned regarding stakeholder engagement for future projects

10.13 Stakeholder Risk Management

- Identifying potential risks related to stakeholder engagement (e.g., misaligned expectations, communication gaps)
- Developing mitigation strategies for identified stakeholder-related risks
- Regular reassessments of stakeholder-related risks as the project progresses

10.14 Communication Tools

- **Project Management Tool**: Task tracking and progress visualization
- **Version Control System**: Code and document version control
- **Documentation Platform**: Centralized documentation
- **Email Client**: For external communications

10.15 Communication Storage and Retrieval

- Storing all project documentation in a cloud-based repository
- Use of consistent file naming convention for easy retrieval
- Maintaining an index of key project documents

12. RISK MANAGEMENT PLAN

11.1 Risk Management Approach

The GrowHub project will adopt a proactive risk management approach, focusing on early identification, ongoing assessment, and practical mitigation strategies. Given the solo nature of the project, emphasis will be placed on manageable risk responses and regular reviews.

11.2 Risk Identification

11.2.1 Risk Categories

- Technical Risks
- Schedule Risks
- Cost Risks
- Quality Risks
- Resource Risks
- External Risks

11.2.2 Risk Identification Techniques

- Review of similar IoT projects and common pitfalls
- Analysis of project constraints and assumptions
- Regular self-assessment of project progress and challenges
- Consultation with online developer communities for insights

11.3 Risk Analysis

11.3.1 Qualitative Risk Analysis

Risks will be assessed based on:

- **Probability**: Low, Medium, High
- **Impact**: Low, Medium, High

11.3.2 Risk Priority

Risks will be prioritized using a simple risk matrix:

Risk	High Impact	Medium Impact	Low Impact
High Probability	Very High	High	Medium
Medium Probability	High	Medium	Low
Low Probability	Medium	Low	Very Low

11.4 Risk Response Planning

11.4.1 Risk Response Strategies

- **Avoiding**: Eliminating the threat or protecting the project from its impact
- **Mitigating**: Reducing the probability and/or impact of the risk
- **Transferring**: Shifting the impact of the risk to a third party
- **Accepting**: Acknowledging the risk and not taking any action unless the risk occurs

11.4.2 Contingency Planning

For high-priority risks, develop simple contingency plans outlining alternative approaches or workarounds.

11.5 Risk Monitoring and Control

- Weekly risk review during Sprint retrospectives
- Maintaining a risk log to track identified risks, their status, and response actions
- Updating risk assessments based on project progress and changing conditions

11.6 Initial Risk Register

Risk ID	Description	Category	Probability	Impact	Priority
R01	Sensor integration challenges	Technical	Medium	High	High
R02	Delays in PCB manufacturing	External	Medium	Medium	Medium
R03	Exceeding procurement budget	Cost	Medium	High	High
R04	Data accuracy issues	Quality	Medium	High	High
R05	Time management challenges	Resource	High	Medium	High
R06	Changes in cloud service pricing	External	Low	Medium	Low
R07	Hardware component shortage or delay	External	Medium	Medium	Medium
R08	Hardware component defects	Resource	Low	High	Medium

Risk ID	Response Strategy	Mitigation Strategy / Contingency Plan
R01	Mitigate	• Starting sensor integration early • Allowing buffer time for troubleshooting
R02	Mitigate	• Ordering PCB early • Having a backup simpler design ready
R03	Mitigate	• Regular budget tracking • Prioritizing essential components • Looking for cost-effective alternatives
R04	Mitigate	• Implementing rigorous sensor calibration process, include data validation in software
R05	Mitigate	• Using time tracking tools • Breaking tasks into smaller, manageable units
R06	Accept	• Monitoring pricing • Having alternative service providers identified
R07	Mitigate	• Ordering critical components early • Identifying backup suppliers
R08	Mitigate	• Prompt inspection upon receipt • Clear quality specifications

11.7 Risk Management Roles and Responsibilities

As GrowHub is a personal project, Filip Vallo will be responsible for all aspects of risk management, including:

- Identifying and analyzing risks
- Developing risk response strategies
- Implementing risk mitigation actions
- Monitoring and controlling risks
- Updating the risk register

11.8 Risk Management Tools and Techniques

- **Risk Register**: Maintained in a project management tool
- **Risk Matrix**: Visual representation of risk priorities
- **Checklist**: List of common risks for regular review
- **Root Cause Analysis**: Understanding and addressing risk sources

11.9 Risk Communication

- Documenting all identified risks and their status in the project documentation
- Including a risk summary in weekly Sprint reviews
- Highlighting any new high-priority risks or significant changes in risk status immediately

11.10 Risk Management Process Review

- Conducting a thorough review of the risk management process at the end of each project phase
- Adjusting risk identification and analysis techniques based on project experience
- Documenting lessons learned regarding risk management for future projects

12. PROCUREMENT MANAGEMENT PLAN

12.1 Procurement Approach

The GrowHub project will adopt a cost-effective procurement approach, focusing on obtaining high-quality components and services within the limited budget. Emphasis will be placed on careful supplier selection, clear specifications, and effective contract management.

12.2 Procurement Items

12.2.1 Development and Management Tools

- Development software licenses or subscriptions (if any required)
- Management software licenses or subscriptions (if any required)

12.2.2 Cloud Services

- Cloud service subscriptions

12.2.3 Hardware Components

- Raspberry Pi microcomputer
- Electronic sensors (temperature, humidity, CO2, moisture, pH, NPK)

12.2.4 Supplies

- Electronic components for peripheral devices
- Electronic components for integrated PCB
- Prototyping supplies (breadboards, wires, etc.)
- Soldering materials

12.2.5 Outsourcing

- PCB manufacturing

12.3 Procurement Process

12.3.1 Identification of Procurement Needs

- Reviewing project requirements and creating a detailed list of necessary items and services
- Prioritizing items based on criticality to project success

12.3.2 Market Research

- Research of potential suppliers and their offerings
- Comparing prices, quality, and delivery times
- Considering both local and online suppliers

12.3.3 Supplier Selection

- For items/services over €100, obtaining at least three quotes
- Evaluating suppliers based on price, quality, reliability, and delivery time
- For critical components, considering supplier's technical support capabilities

12.3.4 Purchasing

- Creating clear, detailed purchase orders or service agreements
- Ensuring all technical specifications are clearly communicated
- Negotiating terms where possible to maximize value

12.3.5 Receipt and Inspection

- Verifying received items against order specifications
- Testing components to ensure they meet required quality standards
- Documenting any issues or defects

13.3.6 Supplier Performance Evaluation

- Tracking supplier performance (delivery time, quality, communication)
- Documenting lessons learned for future procurement decisions

12.4 Procurement Schedule

- Initial procurement of development tools and prototyping supplies: Sprint 1
- Procurement of sensors and main components: Sprint 2
- Cloud services setup: Sprint 3
- PCB manufacturing service: Sprint 5

12.5 Procurement Roles and Responsibilities

As GrowHub is a personal project, Filip Vallo will be responsible for all aspects of procurement management, including:

- Identifying procurement needs
- Researching and selecting suppliers
- Making purchases and managing supplier relationships
- Receiving and inspecting procured items
- Managing contracts and service agreements

12.6 Procurement Documentation

- **Procurement Log**: Spreadsheet tracking all purchases, including item, supplier, cost, and delivery date
- **Supplier Contact List**: Key information for all suppliers
- **Purchase Orders**: Detailed documents for each procurement
- **Contracts/Service Agreements**: For PCB manufacturing and cloud services
- **Inspection Reports**: Documentation of received items and any quality issues

12.7 Contract Management

12.7.1 PCB Manufacturing

- Clearly defining specifications, quality requirements, and delivery timeline
- Including provisions for revisions or issues
- Establishing clear communication channels and points of contact

12.7.2 Cloud Services

- Reviewing service level agreements (SLAs) carefully
- Understanding pricing structure and potential for cost increases
- Ensuring data ownership and privacy terms are favorable

12.8 Procurement Performance Metrics

- Percentage of items delivered on time
- Percentage of items meeting quality standards
- Average time from order to delivery
- Percentage of procurement costs compared to budget

12.9 Ethical and Sustainable Procurement

- Prioritizing suppliers with ethical labor practices where possible
- Considering environmental impact in procurement decisions
- Complying with all relevant regulations and standards

12.10 Procurement Closure

- Ensuring all procured items are accounted for and properly documented
- Closing out any ongoing service agreements or contracts
- Documenting lessons learned from the procurement process

13. CHANGE MANAGEMENT PLAN

13.1 Change Management Approach

The GrowHub project will adopt a flexible yet controlled approach to change management, aligned with Agile principles and the MVP nature of the project. The focus will be on rapid evaluation of potential changes, clear decision-making, and efficient implementation of approved changes.

13.2 Definitions of Change

Changes that require formal management include:

- **Scope changes**: Additions, removals, or significant modifications to project deliverables
- **Schedule changes**: Alterations that impact Sprint timelines or overall project duration
- **Budget changes**: Modifications that affect the overall project budget or significant reallocations
- **Quality changes**: Alterations to quality standards or testing processes
- **Risk profile changes**: New risks or significant changes to existing risk assessments

13.3 Change Management Process

13.3.1 Change Identification

- Changes may be identified through daily Scrums, Sprint reviews, or ongoing development work
- Documenting the proposed change, its rationale, and potential impacts

13.3.2 Change Evaluation

- Assessing the change impact on project constraints (scope, schedule, cost, quality)
- Evaluating the change against project goals and MVP requirements
- Considering alternatives or modifications to the proposed change

13.3.3 Change Approval

- **For minor changes**: Quick self-approval process if the change aligns with MVP goals and has minimal impact
- **For major changes**: Conducting a thorough self-review, possibly seeking input from online developer communities or mentors

13.3.4 Change Implementation

- Updating relevant project documents (e.g., Project Plan, Sprint Backlog, Risk Register)
- Adjusting Sprint plans or project schedule as necessary
- Implementing the change in the development process

13.3.5 Change Monitoring

- Tracking the implementation of the change
- Assessing the actual impact versus the anticipated impact
- Documenting lessons learned from the change process

13.4 Change Tools and Templates

- **Change Request Form**: Digital template for documenting proposed changes
- **Change Log**: Project management tool section for tracking changes
- **Impact Assessment Matrix**: Tool for evaluating change impacts across project constraints

13.5 Change Request Form

Creating simple change request form including:

- Change ID
- Description of the change
- Rationale for the change
- Impact assessment (scope, schedule, cost, quality)
- Alternatives considered
- Decision (approved / rejected / deferred)
- Implementation plan

13.6 Change Log

Maintaining a change log to track all change requests and their status, including:

- Change ID
- Description
- Date submitted
- Status (under review / approved / rejected / implemented)
- Impact summary
- Decision date

13.7 Decision-Making Criteria

Considering the following when evaluating changes:

- Alignment with MVP goals and project objectives
- Impact on critical path activities
- Cost-benefit analysis
- Risk introduction or mitigation
- Technical feasibility
- Impact on overall project timeline

13.8 Emergency Changes

For urgent changes that require immediate action:

- Implementing a streamlined approval process
- Documenting the change and rationale as soon as possible
- Conducting a post-implementation review to assess impact and lessons learned

13.9 Change Management Roles and Responsibilities

As GrowHub is a personal project, Filip Vallo will be responsible for all aspects of change management, including:

- Identifying potential changes
- Evaluating change impacts
- Making approval decisions
- Implementing and monitoring changes
- Updating project documentation

13.10 Communication of Changes

- Documenting all approved changes in the project documentation
- Reviewing implemented changes during Sprint Reviews
- Updating any external stakeholders if changes affect their work

13.11 Measuring Change Management Effectiveness

- Tracking the number of approved vs. rejected changes
- Monitoring the impact of implemented changes on project progress
- Assessing the accuracy of change impact predictions
- Reviewing the efficiency of the change management process during Sprint Retrospectives

13.12 Continuous Improvement

- Regular reviews of the change management process for efficiency and effectiveness
- Adjusting the process as needed based on project experiences
- Documenting lessons learned regarding change management for future projects

14. CONFIGURATION MANAGEMENT PLAN

14.1 Configuration Management Approach

The GrowHub project will adopt a comprehensive yet streamlined configuration management approach, addressing both hardware and software components. The focus will be on maintaining clear versioning, ensuring traceability, and facilitating efficient development and potential future scaling of the MVP.

14.2 Configuration Items (CIs)

The following items will be under configuration management:

14.2.1 Software CIs

- Source code (Python, JavaScript, HTML, CSS)
- Database schemas
- API specifications
- Software libraries and dependencies
- Firmware for microcontrollers

14.2.2 Hardware CIs

- Hardware component list and specifications
- PCB design files
- Wiring diagrams

14.2.3 Documentation CIs

- Project Management Plan and all subsidiary plans
- Technical specifications
- User manuals
- Testing documentation

14.2.4 Development Environment CIs

- Development tools configurations
- Build scripts
- Deployment scripts

14.3 Version Control

14.3.1 Software Version Control

- Source code version control
- Implementing a branching strategy
- Using semantic versioning (MAJOR.MINOR.PATCH) for releases

14.3.2 Hardware Version Control

- Using version numbers for PCB designs
- Maintaining a changelog for hardware components

14.3.3 Documentation Version Control

- Storing documentation in repository alongside code when possible
- Using clear version numbering for standalone documents

14.4 Configuration Management Process

14.4.1 Identification

- Assigning unique identifiers to all CIs
- Using clear naming conventions for files and components

14.4.2 Control

- Implementing access controls
- Using code self-review
- Protecting main/master branch from direct commits

14.4.3 Status Accounting

- Maintain a configuration status log
- Recording the status of CIs (development / testing / released)

14.4.4 Auditing and Review

- Conducting regular self-audits of CIs
- Reviews of CI status during Sprint Retrospectives

14.5 Configuration Management Tools

- Integrated development environment (IDE)
- Source code version control
- Repository hosting source code
- Spreadsheet or database for hardware component tracking

14.6 Baseline Management

- Establishing baselines at key project milestones
- Tagging releases in code repository
- Archiving stable versions of hardware designs

14.7 Release Management

- Defining a release process for both software and hardware components
- Using a consistent naming convention for releases
- Maintaining a release notes document

14.8 Environment Configuration

- Documenting development environment setup
- Using virtual environments for Python development
- Containerizing the application if applicable (e.g., using Docker)

14.9 Configuration Item Backup

- Implementing regular backups of the code repository
- Backing up hardware design files to a cloud storage service
- Maintaining offline backups of critical documentation

14.10 Change Control in Relation to Configuration Management

- Linking configuration changes to the Change Management Process
- Updating relevant CIs when changes are approved
- Reflecting configuration changes in version numbers and changelogs

14.11 Configuration Management Roles and Responsibilities

As GrowHub is a personal project, Filip Vallo will be responsible for all aspects of configuration management, including:

- Maintaining the version control system
- Conducting configuration audits
- Managing releases and baselines
- Ensuring proper documentation of all configuration changes

14.12 Metrics and Reporting

- Tracking number of commits
- Monitoring frequency of releases
- Assessing time spent on configuration management tasks

14.13 Continuous Improvement

- Regular reviews of the configuration management process for efficiency
- Seeking feedback from online developer communities on CM practices
- Updating the CM plan as the project evolves

15. PERFORMANCE MANAGEMENT BASELINES

15.1 Scope Baseline

The scope baseline consists of the approved project scope statement, Work Breakdown Structure (WBS), and WBS dictionary. For the GrowHub MVP, the high-level scope baseline includes:

15.1.1 IoT Device Prototype

- Raspberry Pi integration
- Sensor array (temperature, humidity, CO2, moisture, pH, NPK)
- Basic data collection and transmission capabilities

15.1.2 Peripheral IoT Devices

- Irrigation control device
- Fertilizer injection device
- Power control device

15.1.3 Integrated PCB

- Prototype of embedded PCB with custom electronic circuit
- Soldered electronic components and connectors for peripheral devices onto the embedded PCB

15.1.4 Web Dashboard

- User interface for data visualization
- Real-time data display
- Historical data tracking
- Alert/notification system
- Integration of peripheral devices with the dashboard
- Environmental control system with automation capabilities for managing growing conditions through peripheral devices

15.1.5 Documentation

- Technical specifications
- User manual
- Testing documentation

15.2 Schedule Baseline

The schedule baseline is derived from the project schedule, which is divided into four main phases, each consisting of two one-week sprints:

- **Phase 1**: 08. 07. 2024 – 21. 07. 2024
  - Sprint 1: 08. 07. 2024 – 14. 07. 2024
  - Sprint 2: 15. 07. 2024 – 21. 07. 2024
- **Phase 2**: 22. 07. 2024 – 04. 08. 2024
  - Sprint 3: 22. 07. 2024 – 28. 07. 2024
  - Sprint 4: 29. 07. 2024 – 04. 08. 2024
- **Phase 3**: 05. 08. 2024 – 18. 08. 2024
  - Sprint 5: 05. 08. 2024 – 11. 07. 2024
  - Sprint 6: 12. 07. 2024 – 18. 09. 2024
- **Phase 4**: 19. 08. 2024 – 01. 09. 2024
  - Sprint 7: 19. 08. 2024 – 25. 08. 2024
  - Sprint 8: 26. 08. 2024 – 01. 09. 2024

15.2.1 Key Milestones

- **30. 06. 2024** - Project scope | Project Charter
- **07. 07. 2024** - Project plan | Design Document | Resource gathering
- **21. 07. 2024** - Phase 1: Prototype of monitoring IoT device with sensors
- **04. 08. 2024** - Phase 2: Web dashboard with monitoring and notification functionalities
- **18. 08. 2024** - Phase 3: Prototypes of peripheral IoT devices and embedded PCB
- **01. 09. 2024** - Phase 4: Environmental control system integrated into web dashboard
- **06. 09. 2024** - Project Summary Report | Close out

Performance against this schedule will be measured weekly during Sprint Reviews.

15.3 Cost Baseline

The cost baseline is established based on the approved project budget of €1500, allocated as follows:

- **Development tools**: €200
- **Cloud services**: €200
- **Hardware components**: €500
- **Supplies**: €200
- **Outsourcing**: €200
- **Miscellaneous**: €200

15.3.1 Planned value (PV) distribution

- **Phase 1** (Prototype of monitoring IoT device with sensors): €400
- **Phase 2** (Web dashboard with monitoring and notification functionalities): €400
- **Phase 3** (Prototypes of peripheral IoT devices and embedded PCB): €600
- **Phase 4** (Environmental control system integrated into web dashboard): €100

15.3.2 Cumulative Planned Value (PV)

- **End of Phase 1**: €400
- **End of Phase 1**: €800
- **End of Phase 1**: €1400
- **End of Phase 1**: €1500

15.4 Performance Measurement Approach

15.4.1 Earned Value Management (EVM)

- Calculating and analyzing Planned Value (PV), Earned Value (EV), and Actual Cost (AC) at the end of each Sprint
- Computing Schedule Performance Index (SPI) and Cost Performance Index (CPI)
- Using Performance Measurement Baseline (PMB) to track overall project performance

15.4.2 Agile Metrics

- **Sprint Burndown**: Tracking task completion within each Sprint
- **Velocity**: Measuring the amount of work completed in each Sprint
- **Sprint Goal Achievement**: Assessing the completion of Sprint objectives

15.4.3 Quality Metrics

- **Defect Density**: Number of defects per unit of work
- **Test Pass Rate**: Percentage of tests passed
- **Customer Satisfaction**: Self-assessment of MVP features against initial requirements

15.4.4 Scope Completion

- **Feature Completion Rate**: Percentage of planned features completed
- **Scope Change Tracking**: Monitoring the number and impact of approved scope changes

15.5 Reporting and Review

- **Daily**: Updating requirement statuses and Sprint burndown chart
- **Weekly**: Conducting Sprint Review and updating EVM metrics
- **End of phase**: Updating project dashboard with performance metrics

15.6 Baseline Maintenance

- Baselines will be reviewed at the end of each Sprint
- Any changes to baselines must go through the formal change control process
- Re-baselining, if required, will be clearly documented with justifications

16. PROJECT CLOSURE PLAN

16.1 Closure Approach

The GrowHub MVP project closure will focus on validating project completion, documenting outcomes, organizing project artifacts, and preparing for potential future phases or stakeholder presentations.

16.2 Completion Criteria

The project will be considered complete when:

- All MVP features as defined in the scope baseline are implemented and tested
- The integrated PCB is functional
- All project documentation is completed and stored in the designated repository
- Final project performance metrics are calculated and analyzed

16.3 Closure Activities

16.3.1 Deliverables Verification

- Review of all project deliverables against the scope baseline
- Conducting final testing of the integrated GrowHub system
- Ensuring all documentation is complete and up-to-date

16.3.2 Documentation Finalization

- Completing the final version of the technical specification
- Finalizing user manual and setup guide
- Updating all project management documents with final status

16.3.3 Project Performance Analysis

- Calculating final Earned Value metrics
- Analyzing overall project performance against baselines
- Preparing a final project performance report

16.3.4 Financial Closure

- Reconciling all project expenses
- Preparing final cost report comparing actual spending to budget
- Closing any outstanding financial matters (e.g., cloud service subscriptions)

16.3.5 Project Repository Organization

- Ensuring all code is properly commented and pushed to the repository
- Organizing hardware design files and documentation
- Creating a clear directory structure for easy navigation

16.3.6 Lessons Learned

- Conducting a final retrospective to identify key learnings
- Documenting challenges faced and solutions implemented
- Recording any process improvements or tools that were particularly effective

16.3.7 Future Recommendations

- Identifying potential areas for future development or improvement
- Documenting any known issues or limitations of the MVP
- Suggesting next steps for potential commercialization or scaling

16.4 Closure Documentation

16.4.1 Project Closure Report

- Summary of project outcomes
- Final performance metrics
- Key achievements and challenges
- Lessons learned
- Future recommendations

16.4.2 Final Project Demonstration

- Prepare a demo script showcasing MVP functionality
- Record a video demonstration if appropriate

16.4.3 Handover Package (for potential future phases)

- System architecture document
- Code repository information
- Development environment setup guide
- Known issues and backlog of potential features

16.5 Project Assets Handling

- Archiving project files and documentation in a secure, accessible location
- Documenting the location and access method for all project assets
- Ensuring all assets are organized for easy retrieval in case of future development

16.6 Stakeholder Notification

- Documenting contact information for key external parties

16.7 Personal Reflection and Skill Assessment

- Reflecting on personal growth and skills developed during the project
- Identifying areas for further learning or improvement
- Considering how this project contributes to long-term career or business goals

16.8 Celebration and Acknowledgment

- Personal celebration to mark the project completion
- Acknowledgment of key support received (e.g., online communities, mentors)

16.9 Post-Project Tasks

- Conducting a final backup of all project data
- Closing or archiving project management tools and channels
- Planning for periodic checks of the MVP system to ensure continued functionality

16.10 Closure Checklist

- Creating a comprehensive closure checklist covering all the above activities
- Using the checklist to ensure no critical closure tasks are overlooked

17. GLOSSARY OF TERMS

- **Application Programming Interface** (**API**): A set of protocols, routines, and tools for building software applications. An API specifies how software components should interact and allows different software systems to communicate with each other.
- **Configuration Item** (**CI**): A uniquely identifiable element within a configuration management system, which can be managed and controlled through the configuration management process. CIs can be components of hardware, software, documentation, or any other essential part of a project that requires management to ensure consistency and quality throughout its lifecycle.
- **Daily Scrum** (Daily Stand-up): Formal short daily event of Scrum framework (typically 15 minutes) held every day of the Sprint. It is a key event in the Scrum framework where the Development Team synchronizes their work and plans for the next 24 hours.
- **Internet of Things** (**IoT**): The network of physical objects embedded with sensors, software, and other technologies that connect and exchange data with other devices and systems over the internet. These objects, or "things," can range from everyday household items to sophisticated industrial tools.
- **Minimum Viable Product** (**MVP**): The most basic version of a product that is still functional and usable by early adopters. It includes only the essential features needed to meet the initial user needs and provide feedback for future development.
- **Printed Circuit Board** (**PCB**) - Flat board made of insulating material, typically fiberglass, with conductive pathways or "traces" etched or printed onto it. These traces connect various electronic components such as resistors, capacitors, and integrated circuits, which are soldered onto the board.
- **Raspberry Pi**: A series of small single-board computers developed in the UK by the Raspberry Pi Foundation. It's widely used in IoT projects due to its low cost, modularity, and open architecture.
- **Sensor**: A device that detects and responds to some type of input from the physical environment. The output is generally a signal that is converted to human-readable display at the sensor location or transmitted electronically over a network for reading or further processing.
- **Scrum**: An agile framework for managing complex projects, most commonly used in software development. It is designed to help teams work together, encouraging collaboration, accountability, and iterative progress toward a well-defined goal. Scrum enables teams to deliver high-value products by dividing the work into short, manageable cycles called Sprints.
- **Sprint**: Fixed time-boxed period (usually 1-4 weeks) of Scrum framework during which a Scrum team works to complete a specific set of work items from the Product Backlog. Each Sprint is a mini-project aimed at creating a usable and potentially releasable increment of the product.

- **Sprint Planning**: Formal event of Scrum framework that kicks off the Sprint. During this meeting, the team defines the work to be completed during the upcoming Sprint.
- **Sprint Retrospective**: Formal event of Scrum framework held after the Sprint Review and before the next Sprint Planning, where the Scrum team reflects on the past Sprint to identify improvements for future Sprints.
- **Sprint Review**: Formal event of Scrum framework held at the end of each Sprint, where the Scrum team and stakeholders inspect the increment of the product developed during the Sprint and discuss feedback and future work.
- **Version Control**: System that records changes to a file or set of files over time so that specific versions can be recalled later. It is essential for managing source code in software development projects.