Greenwatch v2
Project Plan Document

Revision 2

Prepared by

Sharome Burton
Delton Hughes
Victor Marchesi
Calvin Walmer

March 6, 2024

Table of Contents ---------- 2

1. Introduction/Overview

1.1 Purpose of Plan

The purpose of this document is to describe the process, organization, and scheduling of the development activities to take place on the project.

1.2 Scope and Objectives

This section provides the main objectives for the Greenwatch project as well as the scope of the project so that the management and development teams can be on the same page with regards to the goals of the project.

1.2.1 Main Objective

Greenwatch shall provide a system for monitoring and controlling environmental variables within the greenhouse located at Bolin Hall at Midwestern State University. The project will be considered a success once an agent has been set up by the customer (Dr. Timothy Pegg) and is actively monitoring and reporting data back to the web server. The current scope of this project is to improve the front-end of the existing project and add better functionality for the end users.

1.2.2 Plan Overview

This document aims to offer a comprehensive summary, focusing on identifying the target audience, outlining practical applications, and presenting diagrams and scenarios for system development. It will provide detail on both functional and non-functional requirements, address user interface design considerations, and identify potential risks.

1.2.3 Specific goals

There exist many goals for this system. At present, most of the goals are underway but there are still some future goals that have been identified.

1.2.3.1 Completed

- Improved the front-end environment
- User must be able to quickly view environmental data for a given room
- Agents must actively monitor temperature, humidity, pressure, and light intensity within rooms.
- The agents which run the monitoring equipment must be able to run on low-cost, Linux-based microcomputers.
- Users must be able to utilize the system via a user interface hosted on the internet.
- Agent must be able to run indefinitely in order to track data for experiments that take place over long periods of time

1.2.3.2 Work in progress

- Greenhouse environments must be monitored using equipment stationed within the physical rooms.
- Agents must be able to take measurements at intervals specified by the user.
- Agents must be able to restart themselves and continue operating in the event of a power disruption, or any other interruption, preserving continuity of the experiments.
- Project must be iterable in order to allow future developers to add features and scale to the system delivered to the customer.

1.2.4 Identified Future Goals

Once phase 1 has been completed we plan on adding additional features to the system, the order in which these will be implemented has yet to be decided.

- Implement different gas sensors (CO2 and Methane) for the agents
- Push notifications based on user specifications
- Vent Control (on/off)
- Shade control (on/off)

1.3 Project Phases

The project will take place in 3 main phases with a 4th phase left to deliver additional features not specified in the success criteria.

Phase 1 - Planning
Phase 2 - Front-End
Phase 3 - Back-End
Phase 4 - Server/Connectivity

- Hosting of server and deployment

Phase 5 - Additional Features

- Push Notifications / Experiments
- Vent control
- Fan control

2. Staff Organization

2.1 Team Structure

The team structure that will be utilized throughout all phases of the project will be a democratic decentralized model. Although Calvin has been designated as the team leader, all of the team's decisions are made through a weekly brainstorming session as well as in continuous communication through discord and text messaging. Because Sharome and Delton are adept at back-end and front-end design specifically, they each have been assigned as the primary developer for each of those subsystems. Any time there is a need for additional manpower to be employed for one of the subsystems either Delton or Sharome can pull in additional resources to finish the task. Calvin and Victor will be responsible for planning and researching systems for deployment in the future as well as doing a majority of the documentation with consultation from Delton and Sharome at the weekly meetings. Calvin and Victor will also work together on setting up the agents and streamlining the process for the customer.

2.2 Management Reporting

Weekly meetings take place on Tuesday in the Dillard 315 Lab at 7:00 PM. At these meetings the team gets together and shares what they have accomplished during the week and actions are assigned for next week. Additionally at these meetings manpower is redistributed depending on the urgency/size of the action at hand. Customer meetings take place each Friday at 5:15 PM in the Pierce common area. At these meetings the development team reports their progress to, and receives feedback from, the customer.

2.3 Communication Methods

The development team will use Discord, text messaging, Trello, and Figma to communicate ideas and stay on track with weekly goals. The primary method that the development team will use to communicate with the advisory team will be email. Should email prove to be too limited, an in-person meeting will be scheduled for the teams to convene.

2.4 Management and Organization

After evaluating various process models, the team determined that the Rational Unified Process (RUP) closely aligned with the input parameters provided by the process modeler from D2L. The compatibility of RUP with our specific requirements, as outlined by the D2L modeler, made it the most suitable choice for our project.

2.4.1 People

The Greenwatch v2 team consists of four members who are actively involved in the development of the system. The advisory team consists of two members; one of whom will be the end user of the system.

2.4.2 Development Team

The development team members and their specialities are as follows:

- Sharome Burton (Back-End Design, Microcomputers)
- Delton Hughes (Front-End Design and Planning)
- Victor Marchesi (OS, Front-End Design)
- Calvin Walmer (Planning, Microcomputers)

2.4.3 Advisory Team

The advisory team members and their roles are as follows:

- Dr. Timothy Pegg (Customer and end user of the system)
- Dr. Catherine Stringfellow (Software engineering expert)

3. Project Resources

3.1 Hardware and Software

Within the entirety of Greenwatch there are two main systems. The first of which being the agents and the code that runs on the agents to report environmental data, and the second being the server that stores the data and provides an interface for users to view data and interact with the agents.

3.1.1 Agent Hardware

Agent hardware consists of a Raspberry Pi 2B that will run the code for taking measurements. The agents do not have the ability to measure environmental data primitively and must have additional hardware that enables this. In addition, the 2B version of the raspberry pi's do not have wifi connectivity. Therefore, a wifi adapter will be required for each agent. A complete list of all agent hardware follows:

- Raspberry Pi 2B, microcomputer on which code runs and measurements are taken
- Sense HAT, allows the agent to measure pressure, humidity, and temperature.
- Photoresistor, measures light intensity through a decrease in resistance of the component
- MCP3008 ADC microchip, converts analog resistance value to a digital signal interpretable by the Pi
- 10 Ω Resistor
- 1 30 row, 8 column prototyping breadboard
- 25 jumper wires

3.1.2 Agent Software

On the agents, there are some required software components but some can be neglected. The items italicized can be neglected without affecting the operation of the agent. All other items however, must be included. A list of installed software components follows:

- Rasbian OS, a linux distro made specifically for raspberry pis
- *Teamviewer Host, remote support software so maintenance can be performed easily*
- agent<agentNum>.py, a script that will drive the measurement components and send data to the server
- *RTL8811AU wifi adapter driver**
- Python
- The following python libraries
    - spidev
    - sense_hat
    - requests

**If the wifi driver neglected, a wired ethernet connection must replace the previous wifi connectivity*

3.1.3 Server Hardware

Because the server will be hosted on the cloud, it will be difficult to say what kind of hardware it will be hosted on. However, because we will be deploying the server files with Docker to ensure that the server will run on any modern day machine.

3.1.4 Server Software

The software that will be running on the DigitalOcean Droplet will enable users to view and interact with the front end of the system on any device from anywhere. Some of the software, like docker, will be used to ensure that the server can easily be migrated

- Flask
- Docker
- Python
- All python libraries
- Sql-lite
- JavaScript
- React.js

3.2 Special Resources

The development team has been shown the inside of the greenhouse as well as the various systems inside it so that they have an understanding of the environment that will house the agents. Additional information may have to be obtained from the MSU IT department if the server were to ever be hosted locally on the MSU network.

4. Project Estimates

4.1 Historical Data Used for Estimates

The historical data for estimation purposes includes information about the time and effort invested in completing projects during past academic courses. Although this data isn't strictly quantitative, it provides a rough idea of the time and effort needed for specific tasks in this project. These tasks involve mastering new class libraries, troubleshooting programs of moderate complexity, effectively and thoroughly annotating somewhat advanced source code, and creating additional materials like reports.

4.2 Estimation Techniques

The COCOMO method was used to estimate the effort required for the project. A NASA developed COCOMO calculator was used to this estimate. A utility was also used to count the number of already existing lines of code. Added to this number was the amount of lines we thought we would need to add based on the existing size of the files and the planned feature additions. Using this method a final number of 15 KLOC was arrived at and plugged into the calculator.

Product Attributes	
Required Reliability	1.00 (N)
Database Size	1.16 (VH)
Product Complexity	0.85 (L)
Computer Attributes	
Execution Time Constraint	1.00 (N)
Main Storage Constraint	1.00 (N)
Platform Volatility	0.87 (L)
Computer Turnaround Time	0.87 (L)
Personnel Attributes	
Analyst Capability	1.00 (N)
Applications Experience	1.00 (N)
Programmer Capability	0.86 (H)
Platform Experience	0.90 (H)
Programming Language and Tool Experience	0.95 (H)
Project Attributes	
Modern Programming Practices	0.82 (VH)
Use of Software Tools	0.83 (VH)
Required Development Schedule	1.00 (N)

Figure 1: Fuzzy logic attributes assigned to COCOMO categories

Figure 1 contains the numbers generated from selecting from Very Low, Low, Nominal, H, Very High, and Extremely High on the calculator. These values were then used by the calculator to produce an approximate time estimate in person-months.

4.3 Estimates of Effort and Duration

From the values that were input into the calculator, an effort estimation was produced:

MODE	"A" variable	"B" variable	"C" variable	"D" variable	KLOC
organic	0.8963614544342686	1.05	2.5	0.38	15.000

EFFORT, (in person-months)	DURATION, (in months)	STAFFING, (recommended)
15.395	7.066	2.179

Figure 2: Final results of estimation techniques POTENTIAL RENUMBER

Figure 2 shows the output of the calculator. The uses for the A, B, C, and D variables were not explicitly mentioned in the calculator except for the formula used to calculate the effort while taking into account the lines of code required. This was given as:

effort = a*KLOC^b

To summarize, the calculator established that the required amount of person-months for this project is 15.395, dividing this by the number of people we have working on the project yields:

15.395 person-months / 4 persons = 3.84 months

Which means that we are on track for completion around the end of the semester as long as we work diligently. Additionally, the calculation assumes that we will be writing every line of code which is not the case because a lot of the code has already been written and we need only add to it.

4.4 Estimates of Cost

Costs will come in the form of hosting the server in the cloud. One-time purchases have already been made to allow the agents to connect to wifi and to be able to be powered. The additional Phase 4 materials will be investigated once the primary phase of the project has been completed.

Power supplies for agents (one-time)	$25.00
Wifi adapters (one-time)	$25.00
Additional Phase 4 hardware (one-time)	TBD
Total one-time purchase costs	**$50.00**

greenwat.ch Domain name (yearly)	$15.00
Server hosting with DigitalOcean (yearly)	$48.00
Total yearly operating costs	**$63.00**

5. Risk Management Strategy

5.1 Risk Management Strategy Used

Risk analysis will take place at each Tuesday meeting in order to make sure we are staying within the bounds of our time and to ensure that we are keeping focused on our success condition established in the requirements document.

Risk	Chance	Impact	Strategy
UI too difficult to convert to React	High	Serious	Deploy each page iteratively
Server table unable to be removed in a reasonable amount of time	Medium	Low	Trace the server software and make a decision as to whether or not it will be worth it to do. Otherwise, code around the table
Team member under-utilization	Low	Serious	Weekly meetings and continuous communication to ensure team involvement
Hardware degradation (Humidity, Temperature)	Medium	Serious	Investigate performance of Raspberry Pis in humid and hot environments and place agents accordingly. There is little we can do to remedy this.
Over implementation of features	Medium	Medium	Establish a success condition during phase 1 and do not start on implementation off additional features until that condition is reached
Security concerns for the web server	Medium	Medium	We plan to secure the data being transmitted by using blueprinted routes in flask and jwt’s.
Team member illness	Low	Low	Redistribute manpower to accommodate

6. Tracking and Control Mechanisms

6.1 Quality Assurance Control

Quality assurance will be accomplished by creating detailed documents that describe the way information flows through the systems and how the system works. Continuous testing of the software as well as weekly customer feedback and live demonstrations will ensure that the software works as intended. All members of the group will be responsible for reviewing the code produced by each other member of the team to ensure that it works as intended. Testing will take place in the form of stress testing the server and the agents. We want to make sure the data can be stored in large quantities effectively. Based on the size of the database, we estimate that a year's worth of data for a single agent should not take more than 50 MB of storage.

6.2 Change Management and Control

Should the plan or requirements need a major change, the team will address it at the Tuesday meeting and decide what the best course of action is. At the meeting the team will discuss the potential impacts of the change on cost and time and decide if it is a worthwhile investment to make. Options will be explored until a best solution is reached, the change will be implemented and manpower will be redistributed to accommodate for potential loss of time.

7. Schedule

See attached Gantt chart

8. References

[1] “STRS COCOMO Calculation,” strs.grc.nasa.gov. https://strs.grc.nasa.gov/repository/forms/cocomo-calculation/

[2] BRAT Corporation, “Process Model Advisor.” 2006

[3] C.-H. Kung, *Software Engineering*. New York, NY: McGraw Hill LLC, 2024.

[4] G. Mathers, “gramcracker40/GreenWatch,” *GitHub*, Jan. 26, 2024. https://github.com/gramcracker40/GreenWatch (accessed Jan. 25, 2024).