CTIS-411 SENIOR PROJECT-I

INITIAL PLAN
(IP)

Team 5 - Members

Berk Özdoruk
Erdoğan Yağız Şahin
Oğuzhan Özkan
Ömer Levent Durdalı

Project Supervisor

Dr. Cüneyt Sevgi

Table of Contents

List of Figures

List of Tables

1. Executive Summary

Elixir is an automated smart irrigation and agriculture system that essentially serves two purposes. The first purpose of Elixir is to give agricultural support and analyze the collected data by providing real-time and historical weather and agriculture data along with visualizations in the user dashboard. Based on the collected and analyzed data, recommendations based on the user's crop type will be presented. The second purpose is to control the irrigation based on user preferences and the data collected from the sensor and external sources. Elixir will provide the user with optimal irrigation recommendations, managing irrigation by itself. The user will also have the choice to schedule irrigation at a specific time or manually start/stop the irrigation. The proposed solution offers more feedback and actionable intelligence to the end-user compared to the competitors in the field.

To have decreased latency, more efficient traffic usage, faster message transmission, and less power consumption [1], the data-centric MQTT protocol will be used for the IoT layer of this project. The end-user will interact with the dashboard over the HTTP/WebSocket protocol. The sensor and user data will be kept in different database systems for efficiency purposes.

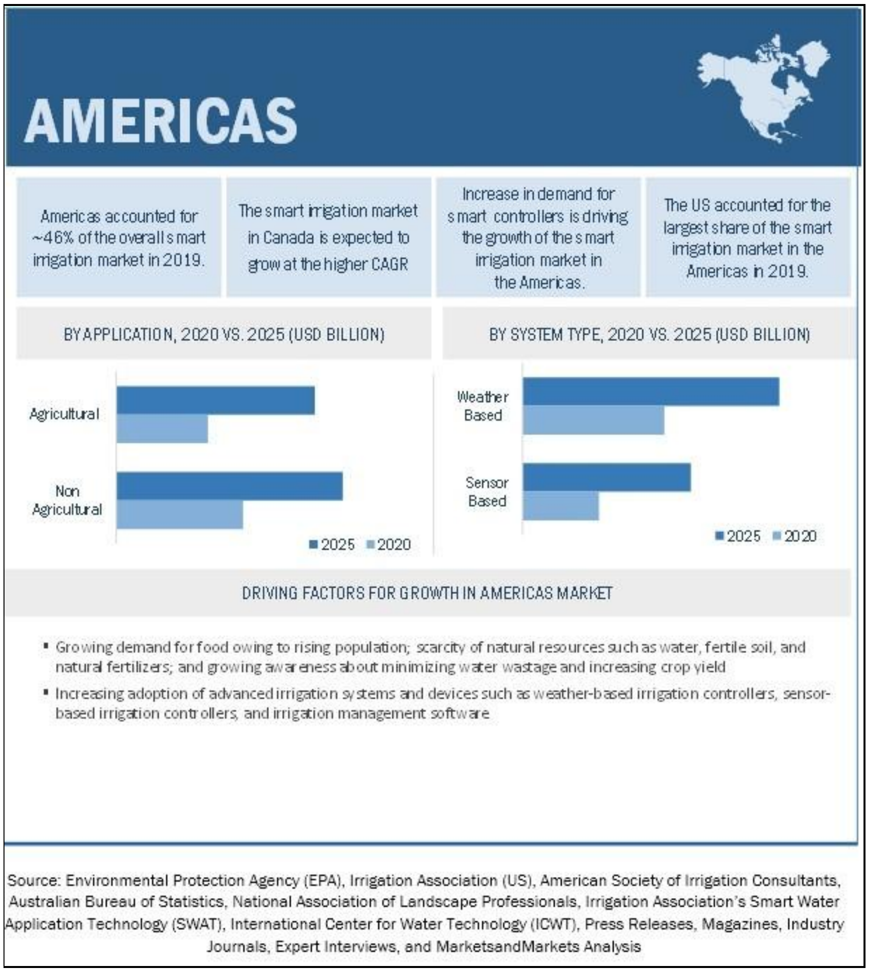
2. Project Purpose

2.1. Problem:

Nowadays traditional irrigated agriculture has some problems mainly because of the huge amount of water consumed. Globally the total fertile area is about 40,000,00 km^2 [2] but only about 3,600,000 km^2 are supported by irrigation infrastructure [3]. Traditional water spring methods use huge amounts of water. Today 2 billion people live in areas where there is water-scarcity [4]. 570 million farms consume farmer's time and effort in the world [5]. Thousands of different types of seeds and each one needs to be raised differently and their number increases rapidly. It causes farmers to be not used to the new methods. Farmers always want their products to grow up in the best possible way. Sometimes they cultivate their field in soils having poor vitamins and minerals. They don't know which plant needs what kind of nutrients. That case has a significant effect on product quality and efficiency.

2.2. Market:

The global smart irrigation market is estimated to be USD 1.0 billion in 2020 and projected to reach USD 2.1 billion by 2025, at a CAGR of 15.3%. [6]
The main reason for the growing smart irrigation market is increasing government encouragement to promote the prevention of wasteful use of water resources.
In the market, there are two types of smart irrigation systems used by farmers. One of them is weather data-based, the other is a soil moisture sensors system. Elixir is able to combine both methods to get the optimal results.


AMERICAS

Americas accounted for ~46% of the overall smart irrigation market in 2019.

The smart irrigation market in Canada is expected to grow at the higher CAGR

Increase in demand for smart controllers is driving the growth of the smart irrigation market in the Americas.

The US accounted for the largest share of the smart irrigation market in the Americas in 2019.

BY APPLICATION, 2020 VS. 2025 (USD BILLION)

BY SYSTEM TYPE, 2020 VS. 2025 (USD BILLION)

DRIVING FACTORS FOR GROWTH IN AMERICAS MARKET

- Growing demand for food owing to rising population; scarcity of natural resources such as water, fertile soil, and natural fertilizers; and growing awareness about minimizing water wastage and increasing crop yield
- Increasing adoption of advanced irrigation systems and devices such as weather-based irrigation controllers, sensor-based irrigation controllers, and irrigation management software

Source: Environmental Protection Agency (EPA), Irrigation Association (US), American Society of Irrigation Consultants, Australian Bureau of Statistics, National Association of Landscape Professionals, Irrigation Association's Smart Water Application Technology (SWAT), International Center for Water Technology (ICWT), Press Releases, Magazines, Industry Journals, Expert Interviews, and MarketsandMarkets Analysis

Figure 1: American market for the smart irrigation [6]

2.3. Elixir and Other Products:

Elixir changes the farmer position to the observer rather than the worker, taking the responsibility that falls on the farmer. The system approaches by using current technology and knowledge as a farmer. Elixir has a wide range of information about that. It can easily guide farmers and provide maximum efficiency. Elixir is able to distribute manure, minerals, and vitamins over a field with pipes. Elixir provides people with the opportunity to save time and effort. It provides irrigation at any time during the day regardless of the wind speed. The system consumes less energy as irrigation is provided at low pressures. The new generation method uses smart technologies, for example, IoT, Big Data, and a wide range of smart devices. It observes and collects data from the field and analyses them for accurate decision making. Smart devices provide people with information that helps them achieve the things that they want to do. Elixir is able to make decisions for them and it can work without human interaction for an extended period of time. Users will be able to watch graphical data, live information about their field, weather status, next watering time, and information about passed watering times.

2.4. What is the COVID-19 effect on the smart water system?

Due to the COVID-19 pandemic, the sector has only suffered a small drop because development restrictions and lock-in have caused interference in the flexible chain. In any case, the use of remote detection and water supply systems integrated into programming tools may prompt the rising period after the COVID-19. The coronavirus has disrupted the flexible chain in the savvy water system market, and organizations are studying new opportunities to use advanced technology to connect with producers and ranchers. It is the chance for increasing the market size for smart irrigation systems because the last period shows that remote systems are suitable for that kind of lifestyle.

3. Project Scope

3.1. Requirement Elicitation:

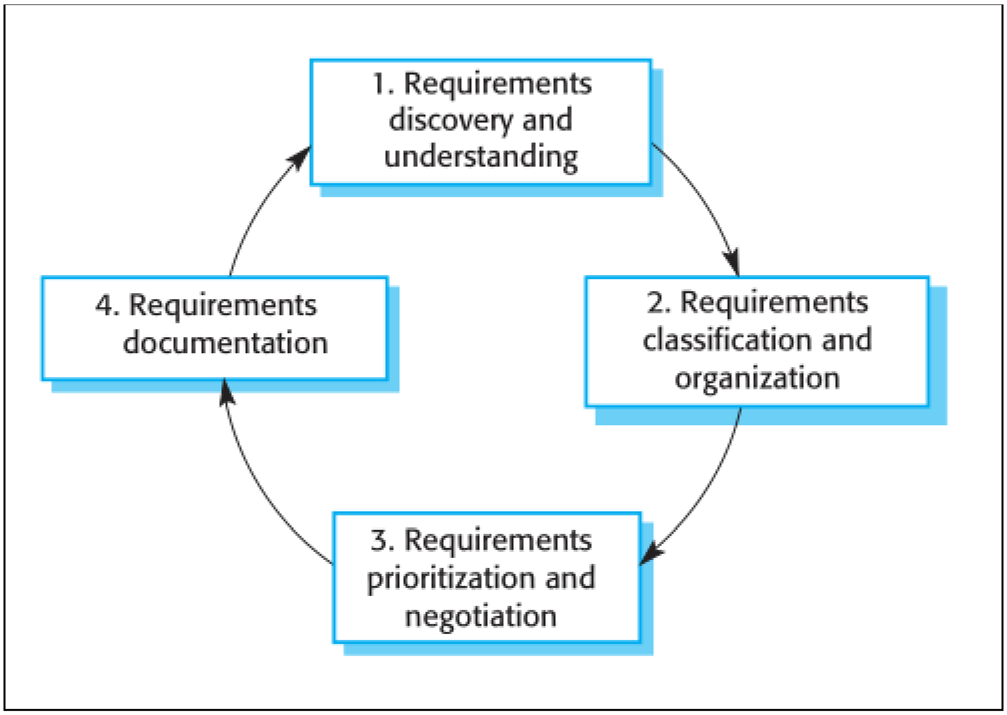

1. Requirements discovery and understanding
2. Requirements classification and organization
3. Requirements prioritization and negotiation
4. Requirements documentation

Figure 2: Requirement Elicitation Cycle [7], [8]

3.1.1. Initial end-user requirements discovery and forming a simple requirements document

- With the help of our advisors and the analysis of existing products, we will decide upon which features would be possible for our project. Additionally, this will allow us to determine which features should receive priority, be simplified, be added, or be eliminated from our initial project requirements.

3.1.2. Extensive online research and Incorporating expert opinion

- In this phase, articles and researches will be analyzed to utilize cutting edge research and combine this knowledge with the expertise of our agriculture advisors. Advisors will be consulted in order to refine the requirements. This is needed simply because of our unfamiliarity with the field. In the end, we will have a list of ideas and tasks to implement, eliminate, or simplify.

3.1.3. Compilation and prioritization

- Finally, all the requirements and tasks will be compiled and they will be assigned priority relative to each other.

3.2. Architecture Design and Development

3.2.1. Development and Integrate services of the project

- Firstly, we have to set up a cloud web service environment like AWS, Azure, etc. along with two databases for sensor and user data respectively. For our Sensor data, we plan on using PostgreSQL (a relational database) or InfluxDB (a NoSQL database) and for our User data, the initial plan is using MongoDB which is also a NoSQL database.
- Additionally to these services we will establish an MQTT workplace for our IoT device so that they can connect and communicate with our server. Once these kinds of backbone-like software are implemented and are verified to be working. We will integrate our initial prototype to fully test our architecture. Then we shall use iterative cycles during the development of our software and hardware as there are many parts that can fail in architecture.
- Currently, no specific libraries and services have been chosen for the architecture these will be decided upon

3.2.2. Testing and Verification

- The software verification and testing will be carried out by each of our team members. Additionally, each member is responsible for testing their own parts of the architecture. This way our team aims to minimize the chance of faults in the design and architecture.

4. Product Requirements

4.1. Users of the product

4.1.1. Farmers

- It is always a problem for farmers to sprinkle all the fields correctly, on time, and be sure about satisfactoriness. Thanks to Elixir, farmers will not have to go to the farm or fields just to sprinkle it. The system will take care of the field for them, and be sure everything goes perfect. Elixir analyzes the crop, knows what that crop needs, and gets hourly live data about the field and shows it to farmers via the Web page or apps. Ensuring your soil is rich, thicker roots flourish, making the most of every drop of water and available nutrients.

4.1.2. Yard and garden owners

- Yard and garden owners always want to see beautiful, sustainable areas. Healthy soil and well-watered plants are the engines behind the success. Elixir takes yard care to a new level where owners have stronger, richer yards and gardens with more efficiency and fruitfulness. Our product will ensure smart watering with correct times, medicines, and with a sufficient amount of water. Thanks to a water smarter system, you can automate watering for better accuracy and efficiency.

4.1.3. Municipalities and Districts

- Municipalities and districts are responsible for very large areas full of trees, flowers, and lawns. Elixir allows you to control large areas like this. A smart, well timing and the correct watering system prevent you from spending a lot of money from maintenance costs to make it green again. Dried up trees, flowers can be a huge load for municipalities and districts. A beautiful green area for today, and tomorrow. Sustainability is the key and we believe that it is easy with the Elixir for a brighter future.

4.2. Functional Requirements

4.2.1. Personal Accounts for users

- Users will have their own private accounts on the website to keep track of their green areas. They will be able to see graphs, live information about their area, current and future weather status, next watering time, optimal condition of seeds, and information about passed watering times.

4.2.2. Visualizations

- According to the user’s area, Elixir understands the conditions of products and will create different types of visual graphs for better understanding abilities and clarity.

4.2.3. Save reports and generate reports

- Elixir will create automatic reports after every irrigation event and in a regular time-based interval for users to see their progress in different formats. (jpeg, png, etc.) In these reports, users can see how much water they spend, what agricultural medicines they used, how many times the system ran and more info about their areas. Optionally, farmers can set the system to auto email the reports to a manager.

4.2.4. Irrigation Control

- Elixir will control the irrigation based on user preferences. A user can initiate the irrigation in real-time using the dashboard. Besides this, the user can set the irrigation on a specific time period or use the pre-defined optimal settings.

4.3. Non- Functional Requirements

4.3.1. Usability

- Data will be gathered by sim-card in the fields and a user will be able to start or stop watering, access all the data and information via the Web site and mobile application. These features will help users about how to use the Elixir with instructions.

4.3.2. Performance

- All data will be created in a short time and no-one should not wait for a long time. The interface should be fast and quick to respond.

4.3.3. Scalability

- Our system, software, and hardware are ready to handle the influx of demand, trends, changing needs, and even the presence or introduction of new competitors. Scalability is important to get lower maintenance costs, better user experience, higher agility, and maintaining effective performance during or after a steep increase in the workload.

4.3.4. Reliability

- Web servers and database servers should be online and can be reachable almost all the time for being reliable. Users should be able to access what they need everywhere and every time.

4.3.5. Security

- The security process will be continuous as new threats emerge and new solutions are needed. We will spend time on penetration testing, system hardening, and source code analysis. Detected vulnerabilities will be patched according to their importance based on severity and priority. User passwords will be hashed using bcrypt along with a salt to limit the exposed information in case of a breach. Information relevant to security will be logged.

4.3.6. Flexibility

- Elixir Web page should work on all smart devices like phones, tablets, and computers no matter the platform. The latest and most reliable technologies must be used to keep up to date. Mobile applications can be released when it is needed.

4.4. Limitations

- Farming areas and fields are all around the world with different corps, fruits, or vegetables. It brings a lot of possibilities and difficulties which can be unrecognizable. A huge amount of effort and detailed scope is needed to succeed in these issues. Sensor numbers and types are limited and we need to consider the efficiency/price of the sensors. Simply, products can be more advanced when you add more sensors or when you have more data. So, we need to decide on the right and adequate combinations of hardware and data.

4.5. Context Diagram

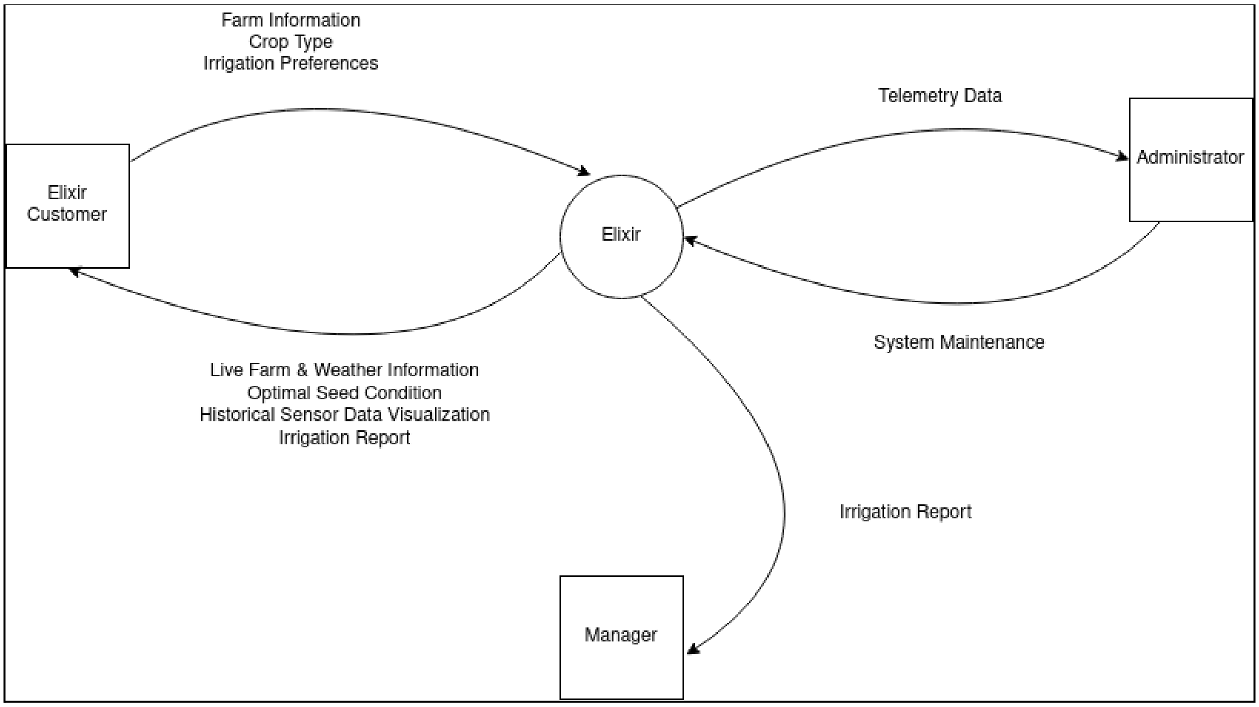


Figure 3: Elixir Context Diagram

5. Software Development Process Model

The agile software development process model will be used for the development of this project.

- Considering that the developers of the project do not have much experience in the IoT and the agriculture field, changes in the project requirements are a big possibility. Due to the frequency of the increments, new changes can be implemented at very little cost in the agile model.
- Due to the nature of the agile model, thorough planning is not required (except the deployment dates). Which will allow us to start quickly, saving time.

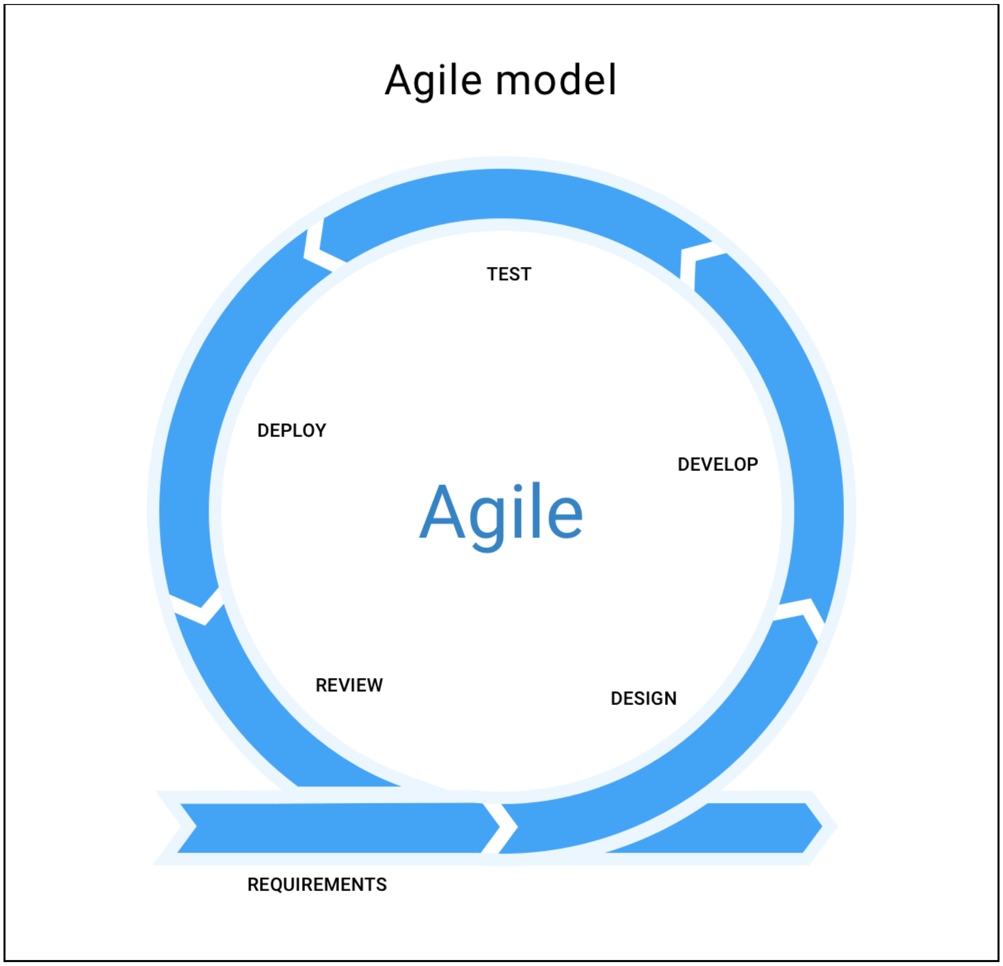


Figure 4: Agile SDLC Model [9]

6. Project Stakeholders and Organization

Stakeholders of the project are Berk Özdoruk, Erdoğan Yağız Şahin, Oğuzhan Özkan, Ömer Levent Durdalı, Dr. Cüneyt Sevgi and Beyhan Akporay. While these people are the main stakeholders. Additionally, Elixir has two domain specialists that will provide help in hardware and agricultural topics respectively. Besides our project consultant (Beyhan Akporay) and advisor (Dr. Cüneyt Sevgi), the roles of the core team members will be interchangeable throughout the project as most of our skills coincide. However, some members will additionally have distinct roles, these roles are as follows: Berk Özdoruk will be the main contact person with the specialists and will also act as the main project organizer, Erdoğan Yağız Şahin will be the main contact person with our project advisor Dr. Cüneyt Sevgi and our project consultant Beyhan Akporay. Additionally, all team members are obligated to complete tasks such as testing, designing, coding, and graphics implementation for the project.

7. Project Communication

Our communication system will mainly consist of real-time communication between stakeholders. WhatsApp and Zoom conference calls are going to be our main points of contact.

We have a WhatsApp group with our specialists, so we can communicate instantly all the time.

We will arrange meetings twice a week at minimum to discuss our progress, implementation details/ideas, and discuss future workflow.

7.1. Project Collaboration:

- Will be done by GitHub Projects. It will allow us to keep track of tasks in different stages.
- The stages will include a list of to-do tasks, in-progress tasks, tasks in the bug-fixing stage, and completed tasks. All these tasks will be divided into Hardware and Software tasks respectively.
- This will also allow us to set deadlines and form a priority list.
- As we have specialists. Using GitHub Projects will make it easier to invite outside members to view the product status and observe our workflow.

8. Project Change Control

In our project, we have a 5-step change control process and this process ensures that change proposed during a project is adequately defined, reviewed, and approved before implementation.

8.1. Proposing a Change

- Anyone in the project team including our advisors and specialists or clients can suggest a change in the project. The suggestion or proposal should have a description of the change and why it is needed.

8.2. Summary of Impact

- Whoever is responsible for a part that has a suggestion of change will consider the overall results of change and need to cover benefits, savings, cost of change, timescale changes, extra resources needed, and new risks or issues.

8.3. Decision

- All group members, with the result of the second process, can approve or decline the change. This decision can be Accepted, Rejected, and accepted or rejected with special conditions- precautions.

8.4. Implementing

- If the change is approved by the people responsible for that specific part, necessary activities must be carried out and be implemented by the team. All acceptance tests must be done in order to avoid getting new errors due to this change.

8.5. Version Control

- All programming phases will have Version Control using GitHub's Version Control System. This will allow us to track changes and issues in the project. It will also allow us to work in sync on a specified repository and make sure there are no conflicts in our code.

9. Milestones & Deliverables

Deliverables/Milestones	Dates
Initial Plan	**Week 4**
Software Requirements Specification (SRS) & Requirements Prototype	**Week 7**
Software Project Management Plan (SPMP) & Gantt Chart	**Week 10**
Software Design Description (SDD)	**Week 13**
1st Increment Product Demo & Project Presentation	**Week 14**

Table 1: Milestones & Deliverables

10. Assumptions

- Cloud services will always be up and running.
- External meteorology APIs will always be up and running.
- Both of the MQTT clients and the broker will be up and running with the capability to restart themselves if the connection fails.
- The MQTT client has to be used between the temperatures of -40 and 85°C [10]

11. Constraints

- All updates will be done via version control systems in order not to lose data.
- A list of security practices will be determined and followed by the developers of the project to prevent API key and source code leaks.
- In order to maintain a persisting code style, standards such as PEP-8 will be followed and team members will be following a predetermined coding style.

12. Risks

Risk	Probability	Affects	Strategy
Incorrect use of agriculture data in the project because of the inexperience with the domain.	Very High	Serious	Research related topics. Consult specialists and advisors regularly.
Group members are ill or unavailable in critical times.	Moderate (COVID-19)	Tolerable	Share out the work to other group members.
Lack of required knowledge and skills.	Very High	Tolerable	Research and learn the required skills in a specified time.
The size of features in the software is underestimated.	Very High	Serious	Determine the most important features. Simplify or eliminate other features and determine the cost of time for these features.
Change in the requirements that require major design rework.	Moderate	Serious	Plan the new workflow. Simplify some requirements.
Free cloud (AWS, Azure, DO, etc.) resources run out.	High	Tolerable, Additional Expenses	Buy additional resources.
Hardware design/development problems because of the inexperience with the domain.	Low	Tolerable	Seek help from project advisors and specialists.
Hardware/Sensor failure due to environmental and unexpected circumstances.	Low	Serious	Find the point of failure, fix or change the specified hardware.

Table 2: Risks for Elixir

References

[1] J. Bartnitsky, "HTTP vs MQTT performance tests", *Flespi.com*, 2018. [Online]. Available: https://flespi.com/blog/http-vs-mqtt-performance-tests. [Accessed: 01- Oct- 2020].

[2] H. Ritchie and M. Roser, "Land Use", *Our World in Data*, 2020. [Online]. Available: https://ourworldindata.org/land-use. [Accessed: 02- Oct- 2020].

[3] "The World Factbook — Central Intelligence Agency", *Web.archive.org*, 2018. [Online]. Available: https://web.archive.org/web/20181214130653/https://www.cia.gov/Library/publications/the-world-factbook/fields/2146.html. [Accessed: 02- Oct- 2020].

[4] "Scarcity | UN-Water", *UN-Water*, 2018. [Online]. Available: https://www.unwater.org/water-facts/scarcity/. [Accessed: 01- Oct- 2020].

[5] "The Number, Size, and Distribution of Farms, Smallholder Farms, and Family Farms Worldwide | FAO", Fao.org, 2016. [Online]. Available: http://www.fao.org/family-farming/detail/en/c/386784/. [Accessed: 01- Oct- 2020].

[6] “Smart Irrigation Market by System Type, Application, Component | COVID-19 Impact Analysis | MarketsandMarketsTM,” www.marketsandmarkets.com, 01-Jun-2020. [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/smart-irrigation-market-199758913.html. [Accessed: 02-Oct-2020].

[7] I. Sommerville, Software engineering, tenth edition, 10th ed. Harlow: Pearson Education, 2019, p. 113.

[8] J. Kondratenko, "Requirements Elicitation: Reasons, Stages, Techniques, and Challenges", Apriorit, 2020. [Online]. Available: https://www.apriorit.com/dev-blog/699-requirement-elicitation. [Accessed: 29- Sep- 2020].

[9] N. Peterheria, "Make Your Startup Work With Agile SDLC Model", Product Tribe, 2020. [Online]. Available: https://producttribe.com/project-management/agile-sdlc-guide. [Accessed: 29- Sep- 2020].

[10] "FAQs - Raspberry Pi Documentation", *Raspberrypi.org*. [Online]. Available: https://www.raspberrypi.org/documentation/faqs/#pi-performance. [Accessed: 02- Oct- 2020].