

ecø, a Rainforest Product

Pablo Acereda García David Emanuel Craciunescu
Pablo Martínez Gracia Laura Pérez Medeiro

October 2019

Introduction

This project will consist of an intelligent system with capabilities to control abiotic factors that affect the wellbeing of plants, with the objective of improving their health and quality of life.

Factors such as humidity and temperature, as well as the levels of light these receive, will be monitorized and analyzed frequently in order to verify and ensure optimal quality of life for the plants.

Thanks to these measurements, a series of alarms and different warning states will be implemented in order to better control the status of the plants and alert the users of their current situation.

Users will be able to easily install the system themselves, and interact and control it via a virtual assistant such as *Amazon Alexa*.

Context

Current situation of the presented problem

The market has seen its fair share of intelligent systems created to aid with plant irrigation control, and there are those that even come with a mobile app interface, such as Blossom, PlantLink or Edyn. Some of the newcomers to jump aboard the backyard-sprinkler train are the so-called ‘intelligent flowerpots’, like the one *Xiaomi* has recently started to sell, capable of watering the plants automatically depending on the humidity of the soil.

There are other projects like *GR0* that are capable of suggesting what plant species one should buy by analyzing the quality and type of soil one uses.

Nevertheless, none of these offers use any kind of virtual-assistant integration or any well-designed user experience, for that matter. *That* will be the main difference our project will have. Not only will the user control the system through a virtual assistant, the design and user experience is planned to be exceptional and extremely easy and intuitive.

End-of-project prediction

Once the course finishes, our intention is to have created a device capable of automatically keeping alive crops or plants with interactions through *Amazon Alexa*.

In order to achieve that, our device will be able to give accurate weather predictions, alert of possible plagues that might attack the crops and monitor the optimal conditions for the plants themselves.

Target audience

This project aims to aid the gardening enthusiast that do not have a great amount of time at their disposal to take care of their plants optimally. It also aims to assist farmers that need help with the care of their crops and seek to minimize the effect of unexpected and external factors to their produce.

Project Scope

Just as previously mentioned, the main objective of this technological endeavor is the creation of a good-enough prototype in order to shot to multiple possible future investors so that they help to commercialize and empower the creation of the product itself through their monetary support. In short and plain English, the objective of the product is to attract suitable investors that, with their contributions, would help the product grow and transform into a marketable commercial system. This product, ‘*ecø*’, is currently the only product the startup *Raninforest* has, therefore this project is of vital importance to the company.

‘*ecø*’ is a product than can be commercialized both as a physical system and a digital solution for the client. As a physical system it will be able to monitor all the ambiental factors that could affect a plant’s growth and development, e.g. humidity, temperature, luminosity, soil quality etc.

As a digital solution, it comes with visualization software that will be able to show and display information in a clear and concise way, as well as in an intuitive manner. The project promoters also hope to be able to motivate a big community of collaborators so that the product improves substantially over time.

This is not all though, for the ‘*ecø*’ also comes with Virtual Assistant integration, with which the user would be able to interact easily and seamlessly.

When it comes to future functions or ones in early stages of development, the biggest one so far has been the incorporation of sensors and mechanisms capable of detecting the presence of bugs and insects that could pose a threat to the plant. There are multiple other initiatives and lines of development and improvement for the product within the company, but those are in very early stages and developing further into those lines of work in this document would not be appropriate, given the volatile nature of initiatives in such early stages.

Discarded Ideas

The project has suffered many changes since its initial conception and ideation. Some of these changes and modifications have been so great and have influenced the project such a big deal that they earn a mention in the following list, just by their very nature as disruptive and polarizing within the developing team:

- **Detection of plagues**

An initial idea was that the system should detect different insects and monitor for known plagues that attack plants and crops. Conceptually insect detection seems simple but its implementation isn't. Some initial ideas entailed the used of infrared radiation, but this idea turned out to be completely inefficient.

This possible feature has not been completely discarded just yet. There is a type of sensor called *PIR*, which could be configured in such a way that detect small-sized animals. The development of this idea is currently experimenting some issues.

- **Development of own microcontroller**

This would have been extremely useful for the product itself, given that the development of a microcontroller specialized for the task would have optimized the way in which the product was designed. After much thought, this idea was discarded, given the short timeframe this product has for development and the ambitious nature of the initiative.

Technology to Use

This section contains the different options that were considered for the project. The different decisions about the different used technology within ght project took into account the experience of the various members of the group, as well as the monetary cost of the different hardware elements, their general availability and the trust the very manufacturer inspired.

Controllers

- **Arduino**

Arduino is an open-source electronics platform that is based on easy-to-use software and hardware. The *Arduino* boards themselves are controlled by sending a set of instructions to a microcontroller on the board. To do so, one must use the *Arduino* programming language, which is based on *Wiring*, and the **Arduino Software (IDE)**, which is based on *Processing*. These boards were our first choice because they are extremely easy to use. The programming language is extremely easy to pick up if one already knows *C*, and the *Arduino* community offers a wide array of free-to-use resources that improve the quality and reduce the effort of any project trying to use *Arduino*.

Even with all its benefits, *Arduino* ended up being discarded as a possible option, given that the absolutely cheapest of boards would still cost around \$20.

- **Raspberry Pi**

According to the official *Raspberry Pi* webpage: “*Raspberry Pi is a low cost, credit-card sized computer that plugs into a monitor or a TV, and uses standard keyboard and mouse. (...) It’s capable of doing what you’d normally expect a regular computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word processing and playing games*”.

Just like a regular computer, one would be able to program it to do whatever they’d want it to do. This would have been ideal for the project itself, given the simplicity of its use and the gigantic array of languages that are compatible with it. In the end, though, it was discarded because the computer itself was too large in comparison to the rest of the elements of the project.

- **ESP32**

The *ESP32* is a series of low-cost, low-power system on a chip **SoC** microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. This was the option the team ended up choosing for the project, given that the *ESP32* was specifically designed for **wearable electronics and IoT applications**.

One can also program on it using the *Arduino IDE*, which was a huge advantage. Most of the team already knew how to use and control *Arduino*, so not having to learn a skill specifically for the microcontroller plus all the benefits of the *Arduino* community made the group decide on the *ESP32* as an option.

Sensors

There are many different kinds of sensors on the market. Many of these are extremely capable and can be obtained very easily. When deciding on the sensors, the team always looked for the most reliable and available on the market. Here are some of the sensors the project will use:

- **DHT22**

It is a basic and low-cost digital temperature and humidity sensor. It measures the surrounding air thanks to a capacity humiditor sensor and a thermistor, and spits out digital signal on the data pin. It’s extremely simple to use, given that its transmissions are already digital, but one can only make readings each two seconds. Extreme speed, or lack thereof, is not a highly important factor when it comes to plant humidity and temperature measuring, so the only real downside would not be an actual downside.

Just like any avid and shrewd reader would have realized by now, this sensor will be used to measure exterior temperature and humidity.

- **LDR**

A *Light Dependent Resistor* or a photo resistor is a device whose resistivity is a function of the incident electromagnetic radiation. These are light sensitive devices also called photo conductors, photo conductive cells or simply photocells.

The ability to detect light is a must in the soon-to-be deliverable product, therefore the use of this sensor was an easy and obvious solution.

- **YL-69**

According to the product description the manufacturer provides: “*The YL-69 is a simple sensor that can be used to detect soil moisture/relative humidity within the soil. The module is able to detect when the soil is too dry or wet. Great for use with automatic plant watering systems*”.

The need for a sensor like this, especially considering it’s insignificant price in comparison to the price of the rest of the setup and components. Therefore, this will be the main triggering mechanism to notify the plant or plants need to be watered.

Programming Languages

As previously mentioned, the chosen microcontroller is the *ESP32*. Therefore, the options when it comes to programming languages have been:

- **MicroPython**

MicroPython an open source *Python* programming language interpreter that is capable of running on small embedded development boards. With the adaptability build into *MicroPython*, one can write clean and simple *Python* code to control hardware directly instead of having to use complex low-level languages.

Even if a reduced version, *MicroPython* still supports most of *Python*’s syntax and implements most of its inner mechanism. Given all these features and advantages, it was a quick initial choice for the project.

These are some of the features that set it apart from other embedded systems:

- **Interactive REPL** This feature allows execution of code without the need of any compilation or uploading time, which is perfect for systems with a high level of experimentation.
- **Extensive software library** *MicroPython* already comes with libraries built in to support common tasks, like JSON data parsing, regular expression handling or even network socket programming.
- **Extensibility** Advanced users may be able to mix *MicroPython* with extensible low-level C/C++ functions in order to further optimize their code and make the execution faster when it really matters.
MicroPython has some very useful features. Unfortunately, it also comes with its downsides:
- Slower code and higher memory needs when compared to C/C++.

- Complicated microcontroller initialization process.
- Limited functionality for some key libraries.
- FreeRTOS

FreeRTOS is a real-time operating system made specifically to run on embedded systems. It is especially good because it natively provides core real time scheduling, inter-task communication, timing and synchronisation primitives. This translates into a more accurately controlled kernel and a system that is able to execute tasks exactly when they have to be executed, a *deterministic* system.

In *FreeRTOS*, applications can be assigned in a static manner while objects themselves can be assigned dynamically with different memory-assignment memory schemes. This system was quickly discarded. Even with the great deal of documentation it counts, the complexity of *FreeRTOS* and the learning curve it would entail made it practically impossible in the provided timeframe.
- Mongoose OS

Mongoose OS is an Internet of Things Firmware Development Framework under Apache License Version 2.0.

Mongoose OS is a firmware development framework specifically designed for development of IoT products. It is highly compatible with a wide array of microcontrollers, just like the *ESP32*, and it's objective is to fill 'the noticeable for embedded software developers' between firmware created for prototyping and bare-metal microcontrollers.

It comes with an integrated web server and supports interaction with both private and public clouds like AWS IoT or Mosquitto.
- Arduino IDE

It's an open-source project that has a very large community behind it. *Arduino IDE* comes with a text editor for writing code, a message area, a console, and other common functionalities of full-fledged IDEs. The programming languages it admits are C and C++, but with some special rules when it comes to structuring the code. The IDE also comes with plenty of built-in libraries for common tasks like robot control or I/O mechanisms.

This is the option the team ended up choosing, not only for its simplicity and usefulness, but also because it can interact directly with the *ESP32* chip that the project will use as a main component.

Development Methodology

One of the key factors when it comes to a project, especially the creation of a new one, is the *Development Methodology*. These methodologies are a means by which a product can be created in an organized, somewhat predictable and planned manner. A *Development Methodology* is nothing more than the process of planning, creating, testing and then deploying a project. And the different

types of methodologies only alter the way in which those components interact with each other.

The specific methodology used for this project is *Lean w/ Kanban*. Here are some of the main reasons that made the group choose this methodology:

1. **Waste reduction.** Helps keep project simple and clean. Everything that is not absolutely necessary is promptly eliminated, be it garbage code, unnecessary meetings, obsolete hardware, etc.
2. **Knowledge improvement.** This is one of the best aspects of *Lean*. Even if every member has a defined function within the team and their tasks are clearly stated, any other member can lend a hand and help if they feel the need or have the time. This helps keep the team agile and on their feet.
3. **Delayed decision-making.** Within *Lean* decisions are made as the requirements are being met, as well as their focus starts to change. The project starts building itself in a progressive and adaptative manner because it hasn't followed a strictly defined idea since its inception. Instead, it has been allowed to grow and morph into it needed to become.
4. **Quick delivery.** Given the minimalism of the methodology and its short iterations of work, the authors of this document believe a further explanation of this specific aspect would be an insult to the reader.
5. **Team-oriented.** Every single member is part of the decision-making process, as well as the definition of their own work. This methodology works best in teams that extremely motivated because the different members usually divide the work amongst them without the need of actual meetings and specific, written-down planification.
6. **Focus on the objective.** Thanks to many of the factors mentioned above, the project does not lose sight of the objective. One could say the *motto* of the team would be "*Think big, act small, err frequently and fix quickly*".

When combining *Lean* with *Kanban*¹ controlling the assignment of different tasks and jobs of the different team members was extremely easy and intuitive. This combination is the key component to the development of the prototype of the product, since it increases visibility of the project and enhances control of the direction in which the development steers towards.

Lean Kanban is a methodology used by various startups, just like IMVU, which is a company created by Eric Ries. He created a methodology called *Lean Startup*, which as 'improved' version of the methodology this project uses, designed specifically to adapt to the way a startup works.

The development cycles in *Lean Kanban* start with the creation of an idea, which is then assessed and developed in order to analyze if its addition to the overall project would be of any real value. With this process, the worse case scenario is still obtaining knowledge about the creation and implementation of that specific idea, even if it does not end up being part of the project. This is

¹ Kanban is a visualization methodology and mechanism, not an actual defined development process.

the way some of the most important parts of ‘*ecø*’ were created, like the *Amazon Alexa* virtual-assistant integration or the detection of factors like humidity and light levels.

Implementation of the methodology

In order to apply the methodology in an effective manner, the team has used the project collaboration tools of *GitHub Projects*, which officially integrated as part of the software of University of Alcalá thanks to the *GitHub Campus Program*.

The team met weekly to plan their activities and tasks for the following week and kept in touch with the mentioned tools, also able to track changes in the status of each task and maintain the visibility of the overall project.

Every single modification and all the planification has been recorded thanks to *GitHub Projects* and a link to the repository will be provided with this document.

Application Architecture

The device is formed by two components:

- Data collector. Formed by all the different sensors that can be found. This component aims to collect and send data to the server. It is the server where the information is going to be treated.
- Actuator. Formed by the irrigation system, it can be activated when the land’s humidity is low or using a voice command.

All the data collected by the sensors is sent to a server, using WiFi and MQTT protocol. The reasons why this protocol has been chosen are, on the one hand, the fact that it is one of the most used protocols in which IoT is referred, and also the fact that we can use OpenMQTTGateway which provided solutions to managed different protocols used by different devices (which means that if there is any sensor that uses BLE, it can be translated into MQTT). There is where C takes part, as it is the language in which *main.cpp* will be created and where all the configuration of MQTT can be found. In the beginning, data is stored into a relational Database which used PostgreSQL as DBMS. Then, data is sent to the AWS DataBase which will be used by Alexa’s API to give the user information about how the plant is.

Also, the provided information by the sensors will be used in a web page, where registered users can find some graphics and statistics.

Business Model

When it comes to selling and pitching the product to customers, there are many factors to take into account. Different elements such as the target demographic or the country in which the product will be commercialized have to be taken into account. In order to be successful, the writers of this document have devised a careful and detailed plan in order to optimize the labors of the company.

There are many things a company must ask itself before selling their products. The first factor of many is about what the company offers, what value do the things that the company does have.

Offering. Value Proposition

Everyone would love to have plants at home without having to worry about how much water they need or whether they are receiving enough sunlight or not. Sometimes people buy plants simply to decorate their homes, but have no idea how to take care of them, or even consider it before buying for that matter. Others may claim to love plants but then have no time, ability or real interest whatsoever when it comes to their wellbeing and maintenance.

‘*ecø*’ can assure their customers that their plants will be well attended at all times. It measures the state of the plant constantly by checking its temperature, humidity, pH levels, etc. After collecting all this information, it carries out a diagnosis, letting its owners know whether the plant is in optimal conditions, whether it’s sick or not², or even if it has any nutrient deficiencies in its soil³.

At the same time, it gives recommendations about the way in which the plant would be better off. In the case of lack of water, ‘*ecø*’ will be able to directly water the plants and, if the plant has other needs, let the owner know immediately.

Furthemore, ‘*ecø*’ has its own skill on *Amazon Alexa*, which makes checking the status of the plants even more convenient. Just by asking *Alexa* about the state of the plant the virtual assistant will give back to the user all the information related to the plant’s current state and the needs it has. If the owner does not have *Alexa* at home, they can also log in to the *ecø* website in order to check all the information related to their plants.

To sum up, the product itself adapts to the needs of each plant, providing its customers with a simple, time-saving, innovative and customised solution. Forgetful or inexperienced customers will not have to worry about possibly killing their plants anymore and plant lovers will have detailed information about the current state of their plants at all times. Combined with the functionalities of *Alexa*, *ecø* offers an even more convenient and efficient way of ensuring the plant’s well-being while taking care of the planet as a whole.

Customers

This section will contain information related to the customers themselves, the point of view the product should have when presented to them, different ‘customer segments’, etc.

Customer segments Nowadays, we all take care of elders, children, animals... But what about plants? They are a key factor in the future of the

² Potential feature in development.

³ Potential feature in development.

planet and the lives of human beings, so there is a general incentive among the population to take care of them. By buying *ecø* this task will be easier and quicker than ever, especially nowadays with faster work and pace of life. However, it is well-known that not everyone is going to be interested in *ecø*. That's the reason why a segmentation strategy must be carried out in order to find the product's targets customers.

By following a segmentation strategy focused on lifestyle, owner's income and individual/business approach one can easily perceive that *ecø* will stick to a niche unattended market formed by people that love plants and have them at home as part of their lifestyles, businesses whose main activity is related to plants or enterprises in sectors such as horticulture, agriculture or gardening. Another option would be targeting public spaces and entities such as governments, public parks, botanic gardens or governmental buildings.

Channels A company can deliver its value proposition to its targeted customers through different channels. Reaching each different segment with different communication strategies is not only important, but vital to the process. Focusing on business-to-business (B2B) and deals with governments is recommended since they normally go hand-in-hand with discounts due to large volume purchases and business negotiations are at stake.

When it comes to sales to clients for individual use, a large range of options are available. In order to attract the attention of those who would be willing to buy *ecø*, the company could promote itself through advertisements on specialized magazines such as 'Country Gardens', 'Fine Gardening', or 'Horticulture', which are read by people interested in plants and probably in *ecø* as well. Another option would be to create online advertisements through gardening forums and blogs, social media such as Facebook groups formed by people interested in the plant worlds, horticulture Instagram accounts, etc.

Furthermore, flyers explaining *ecø*'s benefits and characteristics could be distributed in botanic gardens or plant-specialized stores. Indirect advertisements played on screens during public events related to gardening could be very efficient to find *ecø*'s target customers as well as word-of-mouth among friends, relatives and colleagues who have bought the product and are satisfied with it.

ecø will only have an official distribution channel: online selling, where buyers can directly buy the product online.

Customer Relationships The relationship with the customer does not end once they buy *ecø*, the company will create a phone line and email address so that the customers who are having issues with the product or need information requests can contact the firm directly. Therefore, personal assistance will be available along the whole purchase process, either during pre-sales and/or after sales situations.

It will also create an online community, which allows for direct interactions among the company and its clients, where knowledge can be shared, and problems can be solved between different clients with the company as an intermediary.

Through this platform, consumers could become what is known as co-creators of the product by suggesting ideas of upgrades that would make *ecø* more efficient.

Infrastructure

Key Activities The key activity required to execute *Rainforest's* value proposition are the technical development and programming of the project itself, both in terms of creating a product capable of detecting the plants' state and needs, and creating the website and the Alexa Skill. Another key activity would be to target *ecø's* potential customers correctly by investing appropriately in the different channels needed to create awareness in consumers' minds that, hopefully, will end up in the product's purchase.

Key Resources In order to create *ecø* and sustain *Rainforest's* Business, physical components are needed such as the DHT22 to detect humidity and temperature, microcontrollers and boards such as the ESP-32 and water pumps such as the R385. Apart from that, human capital is crucial for the creation and correct functioning of the software that detects the plant's current state, the creation of *Alexa's* skill and *ecø's* website.

Partner Network Partnerships and alliances are crucial for a business, especially when the company is new to the marketplace. They optimize operations and reduce risks so that the company can focus on its core activity, which in *Rainforest's* case is the creation of *ecø*.

The main and most important partnership for *Rainforest* is the one with the physical distributors of their product. Even if the product is sold online, it still has to arrive to the client in order for them to benefit from it. Choosing the right partners is essential for the company to ensure that quality of product is maintained and the relationship with the client is kept honest and trustworthy.

Finances

Cost structure *ecø's* business structure would be value-driven, since the enterprise will focus its efforts on creating value for its customers rather than minimizing costs as much as possible. It is clear that the company will try to minimize its costs to get higher revenues, but this is not the main objective for the firm at the time this document was being written.

In terms of costs, the key activities that will be more expensive throughout the process are the technical development and computing programming of the project itself, both in terms of the low-level software and the multiple user interfaces. Whenever possible, especially when selling to businesses and governments, economies of scale could be reached, thus, decreasing the cost of each unit as the number of units that are ordered of the product increase.

Revenue Streams *ecø*'s customers would be willing to pay a premium compared to other competitors for the quality and value of the product. *ecø* offers possible new upgrades that are not out in the market yet, such as detecting if the plant is sick or what the disease that is causing it is. Other attributes for which customers would be willing to pay more are its complimentary website and the opportunity of connecting it to *Amazon Alexa*.

Taking a look at the market, the cost of material to create the product would be around \$30US. To this cost, other fixed costs such as the creation of the website, the design and implementation of the physical system itself and the *Alexa* skill, must be added. Comparing it to the market, the most similar competitor would be "Xiaomi Mi Smart Flowerpot Plant Pot" which costs \$50US.

This fact provides *Rainforest* with an idea of what price would be reasonable to ask for the product and still generate revenue, but taking into account that *Xiaomi*'s product doesn't offer as many attributes as *ecø*, it would be reasonable to ask for approximately \$60US as the final price for the product. The revenue streams of the firm will mainly come from the sale of this very product and this price would give *Rainforest* a decent revenue margin to ensure its success in the future.

Marketing

When it comes to actually getting to the clients. There are many factors to take into account. From the way the company is perceived in the market, to the way customers see the product and the sentiment they have towards it. In order to maximize the labors of the company and market the product in the best way possible, *Rainforest*'s marketing team has devised the following factors and elements to follow.

Perception and differentiation When it comes to *Rainforest* itself, the focus and the angle the company should have towards its customers and towards the market could be characterized as that of the personality of a young technological activist that fights for the future of planet Earth. Some of the most important factors this company should have and the way it should behave are the following:

- **Young.** With *young* the team does not mean the company should act childishly, but with the desire to do something and the wish to work hard and earn people's trust.
- **Connected.** Maintaining a connection with the customer, be it through social media or through a different online presence is a key factor when appealing to the younger audiences and customers. The very inclusion of 'technology' in the product will attract younger customers towards it and statistics show a company that keeps in touch is a company customers trust.
- **Revolutionary.** This is not a term that defines the product or the way to do things itself. With a *Rainforest* that is revolutionary what the marketing

team means is a company that is not afraid to do things differently and a company that does not simply fear change for the sake of change.

- **Responsible.** At the same time as young, connected and revolutionary, the company should always maintain its status as a responsible one. The product *Rainforest* sells is one that helps nature and keeps the planet clean. It's a product that helps people reduce wasted resources and keep plants alive, therefore one could say the product is responsible with the environment, and so should the company be.
- **Involved.** An involved and present company is one that is sure to succeed, especially when it is related to nature and the topics *Rainforest* treats and gets involved with. A company that is about the betterment of nature and the improvement of it *must* be one that is involved.
- **Honest.** This point is obvious. A company that does not lie and stays honest with their customers is a company that gains their trust. A company that inspires security and makes people feel confident when buying their products.

Development Plan

Given the very nature of the software development plan the team has chosen for the project, the need for an actual 'development plan' *per se* is not of the utmost importance. The team has instead chosen to divide the work in a realistic set of project objectives and deliverable items within the provided timeframe of the project.

It is true that projects generally function better with a somewhat defined development workflow. This case, however, is not applicable to the engineering team at *Rainforest*. The very nature of their relationship and the way members are able to collaborate seamlessly given their already based background makes the *Lean* software development methodology a great one for the team. In fact, the addition and creation of a software development plan into the development environment would take more time than the actual development itself.

Every single characteristic found in this document and defined as an objective can be considered a deliverable item unless specified otherwise. This section is also accompanied by a document called "planning" where a preliminary idea and conceptualization of how the work should be distributed among the work months can be shown. This, however is a mere conceptualization of a preliminary idea, it should not be taken as definitive.

Risk Assessment and Contingency Plan

All the information related to this point can be found in the file called as risk_analysis-contingency_plan.xlsx

Overview of the Project

Even though the project seems to be seriously ambitious, all the team members are highly motivated and have studied all the relevant information carefully with the objective of having the clearest set of mind by the time the project developing takes place. In order to find collaborator is key to have a well defined project., that is why all the developers have participated in the implementation of the code, developing Alexa's skill, creating the server, etc. Using close-loop feedback within the team members there have been overcome the problems and difficulties faced throughout the implementation of the project.

The aim of this project is, not only to create a system which is able to take care of a plant, but it also to learn and enjoy the process of developing this idea.