Does the Lean Inception Methodology Contribute to the Software Project Initiation Phase?

Igor Braga[1(✉)], Marcelo Nogueira[1,2(✉)], Nuno Santos[2,3(✉)], and Ricardo J. Machado[2,3(✉)]

[1] Software Engineering Research Group, Paulista University, UNIP, Campus Tatuapé, São Paulo, Brazil
marcelo@noginfo.com.br
[2] ALGORITMI Centre, School of Engineering, University of Minho, Guimarães, Portugal
[3] CCG/ZGDV Institute, Guimarães, Portugal

Abstract. This work aims to make a comparative analysis of the Lean Inception methodology and the Scrum methodology applied in the initiation phase of the embedded software project that consists of a small greenhouse for the indoor cultivation of sage, controlled by a cell phone application. The Lean Inception methodology is a combination of Lean Startup and Design Thinking that, at the end of the process, quickly obtains the Minimum Viable Product (MVP). Some of the questions explored in this paper were how much the Lean Inception methodology interferes in the agility of the production of MVP, how much it influences in the quality of the final product and depicting advantages in using this methodology in the project initiation phase.

Keywords: Lean Inception · Scrum · Project initiation · Minimum Viable Product

1 Introduction

Many software projects end up failing due to the lack of good planning, where most of the time they are one or more of the 3 main pillars affected: scope, time and cost.

The solution to this type of problem is a good survey of requirements with the user, the validation of the defined requirements and once every-thing is well defined, the initiation of modeling, because from it the scope, time and cost will be defined.

The importance of modeling is precisely the fact that it is the basis for the definition of the three pillars, and if there is a need for the user to make changes after the definition, we can see how much this change will influence the final delivery of the project.

For a better result, the Lean Inception methodology is applied in the initiation phase of the project, where the diagrams will already be developed through detailed requirements with the user and with that the time, cost and scope will already be

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12254, pp. 741–752, 2020.
https://doi.org/10.1007/978-3-030-58817-5_53

defined culminating in a quality Minimum Viable Product (MVP) that serves the user and there is the possibility of adding value to the software developed over time.

The methodology also aims to develop people, improve work, have a different development change from the project, do until it is interpreted in a more relaxed way, many times this can generate a final product with a good quality.

To improve business results through Lean practice, it is necessary to change not only the processes, operations or include new technologies in the scope of work, but the mindset of the organization as a whole, guiding people to look at the internal factors that involve the company or product development. It is the ability to eliminate waste continuously and solve problems in a systematic way.

Software process it is a set of activities, actions and tasks performed in the creation of some artifact. The view of the activity is an objective without worrying about the difficulty and the effort generated, the action is the set of activities that result in a fundamental software artifact and the task focuses on a small, but well-defined objective. In general, the process is an adaptive approach that allows people to do the job in the best way by choosing the appropriate set of actions so that the software is delivered on time and of sufficient quality to satisfy the people who are sponsoring the development and those who are going to use it [1].

According to The Standish Group there are 4 main reasons for the significant improvement that has been happening over the years since 2010, with the application of agile methodologies being one of the main ones, the use of these processes grows at a rate of 22% CAGR (Compound annual growth rate), today this represents 9% of all IT projects and is adopted in 29% of new application development. The group concludes that the growth in the success rate is directly related to the increase in the adoption of agile methodologies [2].

Another reason is the cascade processes that consist of traditional methods and have already represented almost 50% of the number of new implementations, however, as they grow at 1% CAGR (Compound annual growth rate), their relative use has decreased, thus contributing positively to the success rate [2].

In the project that was chosen for the comparative analysis of this article, two different agile methodologies were applied, the first of which was scrum using simple Kanban and the second was Lean Inception.

The Scrum methodology do not have the part of people development and stimulus of creativity even though it is effective in the part of deliveries on time, however Lean Inception, on the other hand, promotes several techniques, such as people development, creativity stimulation work, through cycles passing through phases where there is the participation of everyone involved in the project, each phase has a different form of brainstorm until reaching a minimum viable product.

Based on studies, it has been proven that in terms of creativity for a minimum viable product for the user, the lean inception methodology has a faster result, mainly due to the fact that the idea is to develop the MVP as quickly as possible and over time add value to the product according to user requests and needs.

2 Background

2.1 Software Engineering

The basis for software engineering is the process layer. The software engineering process is the alloy that keeps the technology layers cohesive and makes it possible to develop software rationally and on time. The process defines a methodology that must be established for the effective delivery of software engineering technology. The software process forms the basis for controlling the management of software projects and establishes the context in which technical methods are applied, derivative products are produced (models, documents, data, reports, forms, etc.), milestones are established, quality is guaranteed and changes are managed appropriately [1].

Process is a set of activities, actions and tasks performed in the creation of some work product (work product). An activity striving to achieve a broad objective (for example, communicating with stakeholders) and is used to apply the field of application, the size of the project, the complexity of cuts or the degree of rigor with software engineering will be applied. An action (for example, architectural design) involves a set of tasks that result in a fundamental software artifact (for example, an architectural design model). A task focuses on a small but well-defined objective (for example, performing a unit test) and produces a tangible result [1].

In the context of software engineering, a process is not a strict prescription for how to develop software. Rather, it is an adaptable approach that allows people (a software team) to do the work of selecting and choosing the set of actions and tasks to be adjusted. The intention is always to offer software on time and with enough quality for the requirements that sponsor its creation and the requirements that will be used [1].

The basics for using any generic process methodology for software engineering comprises five activities:

- **Communication:** Before starting any technical work, it is vitally important to communicate and collaborate with the customer (and other interested parties). The intention is to understand the objectives of the stakeholders for the project and to survey the needs that will help to define the functions and characteristics of the software [1].
- **Planning:** This makes it easier for the team to better understand the progress of the steps. A software project has many steps, and the planning activity creates a "map" that helps guide the team. The software design plan defines the software engineering work, describing the technical tasks to be carried out, the probable risks, the resources that will be required, the resulting products to be produced and a work schedule [1].
- **Modeling:** An "outline" of the software project is created, so that one can get a general idea of the whole. Modeling is used to understand the needs of the software and the design that will meet those needs [1].
- **Construction:** This activity combines code generation and tests necessary to reveal errors in the coding [1].
- **Delivery:** The software is delivered to the customer, who evaluates the delivered product and provides feedback, based on the evaluation [1].

These five generic methodological activities can be used for the development of small and simple programs, for the creation of large applications for the Internet and for the engineering of large and complex computer-based systems. The details of the software process will be very different in each case, but the methodological activities will remain the same [1].

It is based on this information that the ideal methodology for a software project is defined, based on what is required by the project (term, scope and cost) that the ideal methodology to be used is chosen.

In this article, we will approach the two methodologies used in the same project to focus on the difference in modeling used in each of the methodologies, the objective is to make a comparative analysis between the models and the results of the project.

In the next chapters, a brief explanation of each of the two methodologies used in the project is defined.

2.2 Scrum

Scrum (the name is derived from an activity that occurs during a rugby match) is an agile software development method that was conceived by Jeff Sutherland and his development team in the early 1990s. In recent years, further development on the Scrum methods has been performed by Schwaber and Beedle [1].

Scrum principles are consistent with the agile manifesto and are used to guide development activities within a process that incorporates the following framework activities: requirements, analysis, design, evolution, and delivery. Within each framework activity, work tasks occur within a process pattern (discussed in the following paragraph) called a sprint. The work conducted within a sprint (the number of sprints required for each framework activity will vary depending on product complexity and size) is adapted to the problem at hand and is defined and often modified in real time by the Scrum team. The overall flow of the Scrum process is illustrated in Fig. 1.

Scrum focuses on using a set of software process standards that have proven to be effective for projects with tight deadlines, changing requirements, and business critical. Each of these process patterns defines a set of development actions:

The backlog is a prioritized list of project requirements or features that provide value to the customer. Items can be added to that record at any time (this is how changes are made). The product owner evaluates the record and updates priorities as required [1].

Backlog items consist of units of work requested to reach a requirement established in the backlog. Changes - for example, in backlog work items - are not introduced during sprints execution. Therefore, the sprint allows team members to work in a short-term but stable environment [1].

Scrum meetings - these are short meetings (typically 15 min), held daily by the Scrum team. Three key questions are asked and answered by all team members:

- What have you accomplished since the last team meeting?
- What obstacles are you encountering?
- What do you plan to accomplish by the next team meeting?

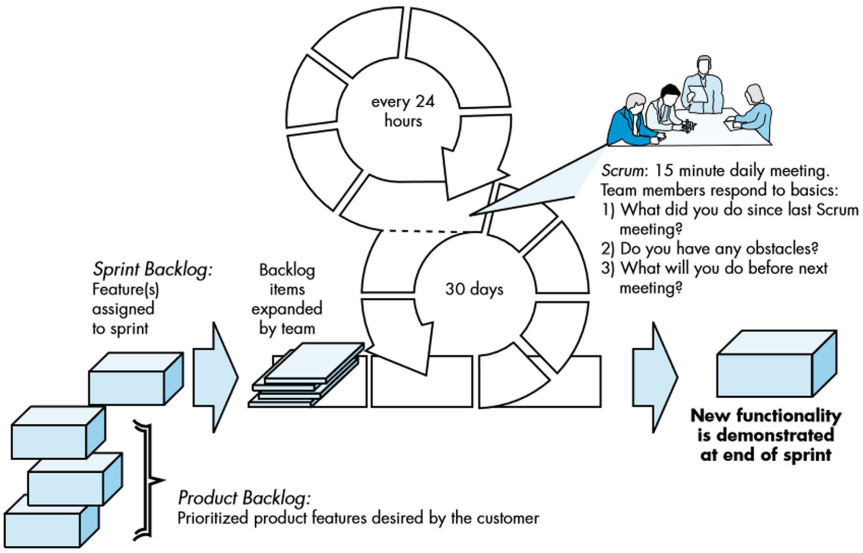

Fig. 1. Scrum process flow [1]

A team leader, called the Scrum master, conducts the meeting and evaluates the responses of each member. The Scrum meeting, held daily, helps the team to reveal potential problems as early as possible. It also leads to "knowledge socialization" and therefore promotes a self-organized team structure [1].

Delivery of the software increment to the customer is performed at the end of each sprint so that the implemented functionality can be demonstrated and evaluated by the customer. It is important to note that the demonstration may not have all the planned functionalities, but functions that can be delivered on time [1].

3 Lean Inception

A "Lean Inception" [4] is useful when the team needs to iteratively develop an MVP. Although the term is often misunderstood, the central property of an MVP is that it is something we build in order to learn whether it is worthwhile to continue building a product. Therefore, we choose features based on testing our assumptions of what is valuable to our users. For this, we need to understand who our users are, what activity they do that the product supports and how to measure if they find the product useful [4].

Lean Inception is valuable in two main circumstances:

- Large projects find a "Lean Inception" valuable to start quickly and be oriented to work in a "Lean" style. Such a start builds early iterations designed to discover and test what features are truly valued by their users [4].
- Smaller organizations (such as startups) use lean inceptions to take an idea that's been tested by some pre-software MVPs and evolve it into a software product [4].

MVP is the simplest version of a product that can be made available to validate a small set of assumptions on the business [4].

Basically, you don't want to waste time, money and effort building a product that won't meet your expectations. For that reason, you need to understand and validate your hypothesis about the business. MVP helps to validate and learn the fastest way [4].

Different from products created using traditional methods, usually taking too much time and effort for prototyping, analysis, and elaboration, the goal of MVP is only to validate the first step – the minimum product – which is far less developed than the final version. MVP focuses on the minimum but viable product to verify if the direction is correct. The initial set of functionalities needed for hypothesis validation and for learning more about the business [4] (Fig. 2).



Fig. 2. MVPs for grass grooming [Source: Lean Inception p. 12]

The focus of lean inception is to create functionalities for an MVP based on the pains of personas with business-oriented modeling, always based on user experience.

4 Case Study

Anxiety disorders have become very common in people's daily lives, especially in Brazil where, according to WHO studies, about 9.3% of the population is affected with some type of anxiety disorder. This rate is considered the highest in the world, and from there we can imagine the weight of this scenario in our country. Some socioeconomic factors such as unemployment, poverty and lifestyles in big cities facilitate the opening to these disorders that can also trigger something more serious, such as depression, among other health problems [5].

Anxiety most often occurs at times considered to have a great emotional impact, which can be classified as "The Evil of the Century" and even as a public health problem, triggering difficulties in concentration, sleep problems and excessive worry [6].

Currently, there are few proven alternatives for the treatment or reduction of symptoms, some are psychotherapy, exercises and medicines, but there is still a lack of natural and usual means of daily life for those who have a running routine and little time for dedication.

The groundwork was created based on three methodological procedures:

- Bibliographic: which involves the analysis and interpretation of documents, articles, manuscripts, etc.
- Exploratory: it is the understanding of a subject little known by the group. At the end of this research, we will have a greater knowledge on the subject, therefore, we will be able to build new hypotheses.
- Experimental: Based on data from articles on existing research, the data regarding what the plant needs were manipulated for the best development of sage and applied this in the greenhouse to propose the exact climatic conditions and with the best quality in the development of the plant.

This was the human cause and the research methods that served as a basis for the development of the idea.

The general objective of the project was the development of a greenhouse with an embedded system in real time, fully adaptable for the best development of the medicinal plant called Salvia Officinalis.

One developed together with embedded software, which makes connection via bluetooth with a smartphone, making it possible to fully control and monitor all functions as the user synchronizes the application with the stove.

The specific objective is to allow the user to have control of the environment in which the salvia will grow, changing the temperature and monitoring according to parameters that will be presented in the application.

It was developed together with embedded software, which makes a Bluetooth connection with the smartphone, making it possible to modify and monitor the internal climate of the greenhouse as the user synchronizes the application, providing healthy development of sage, allowing the user to grow independently and food consumption to help treat anxiety.

The project parameters are:

- Air Humidity
- Temperature
- Water tank
- Ventilation
- Soil Moisture
- Water Pump Operation
- Lighting

The functioning of these parameters influences the internal climatic condition of the greenhouse and they work according to the temperature change, by the user on the smartphone, which is the trigger for the functioning of all components.

There is a manual temperature mode where the user sets the value and the stove works based on what has been defined and there is a default mode, for users who do not have an in-depth knowledge of cultivation, where the stove will operate with the standard settings defined in creation where the temperature value is set to 26° C.

4.1 Project Instance on Scrum

According to the requirements, two use cases were developed, one from an overview and the other from a detailed view for good modeling and understanding of the project (Figs. 3 and 4).

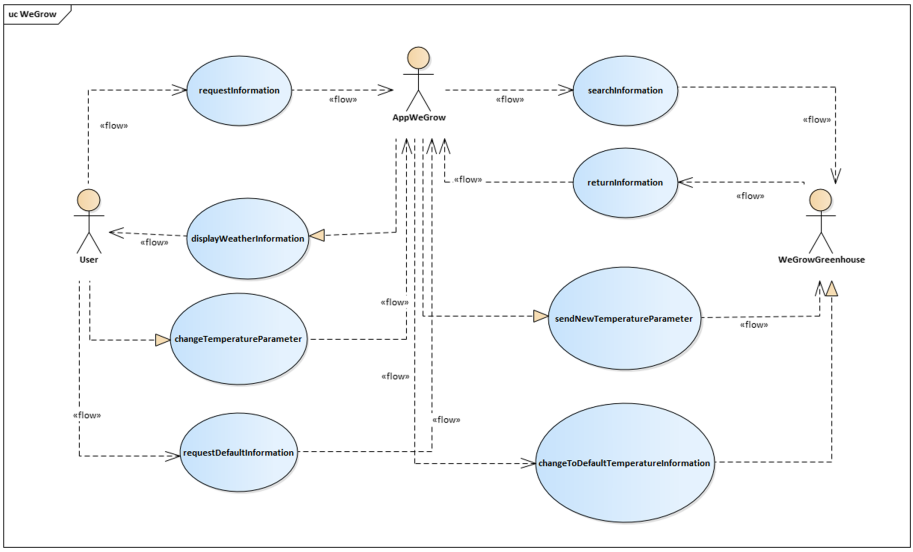

Fig. 3. General UseCase

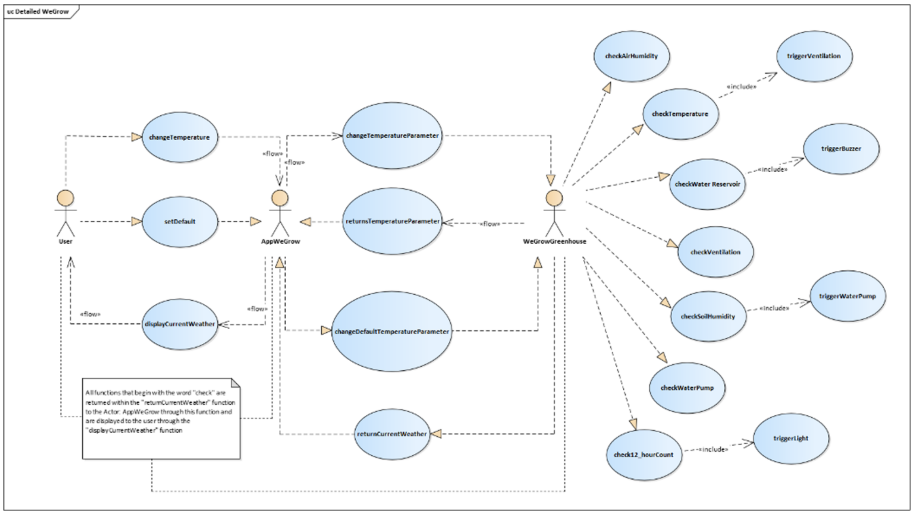

Fig. 4. Detailed UseCase

The methodology for the development of the project was Scrum, where the tasks were divided among the members of the group in such a way that each of them went through the review of all 4 members of the group to finish the sprint, culminating in the completion of each cycle was agile and quality.

In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. This is because the team will know best how to solve the problem they are presented [3].

That's why all team planning meetings were done quickly based on the desired outcome and focusing on features.

As the sprints progress, there is the possibility of extorting bugs, the task identified as a bug is in a kanban column specific to errors, where the programmers assume one by one for the correct correction and progress of the task.

4.2 Project Instance on Lean Inception

The objective of this project was to develop an MVP where the user could feel comfortable using the application and obtain a good result.

The result was a high speed in the delivery of the MVP where the initial result for the user was as expected.

It was identified through the survey of the user's pains primarily in the user's wish not to have healthy cultivation of sage in the external environment due to the interference of climatic variation of the environment.

Based on the project made in the scrum methodology, the idea was also raised that the user could have as an initial MVP an indoor sage cultivation greenhouse that did not have the functionality of modifying parameters manually because it was concluded

that the minimum that the user needs to have a healthy plant is the greenhouse in operation according to the standard configurations, that is, an MVP where the greenhouse would work alone with its factory standards, with the settings and codes set inside the arduino, programmed to work as soon as there is electrical power in the hardware, without user intervention (Fig. 5).

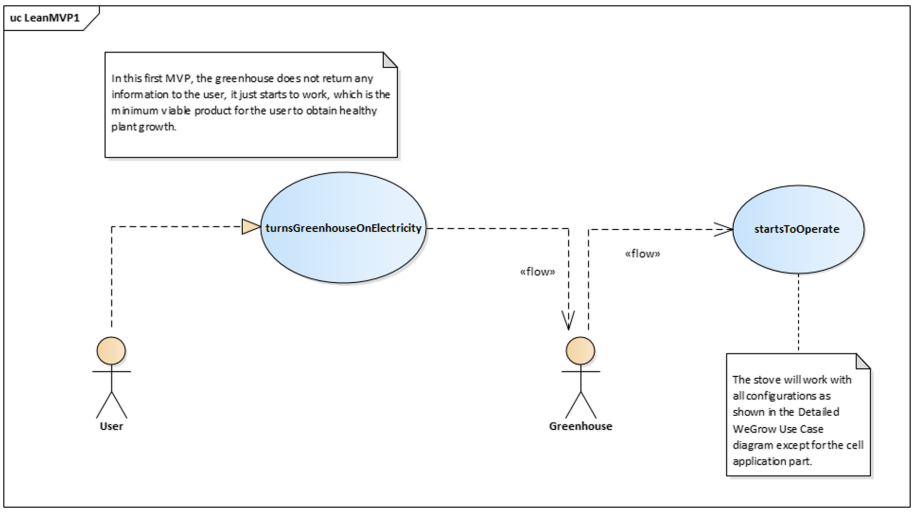

Fig. 5. First MVP UseCase

In a second MVP, the greenhouse would already communicate via Bluetooth with a smartphone, however, only for monitoring, that is, the user would only monitor the information of air humidity, temperature, water tank, ventilation, soil moisture, operation of the water pump and lighting and would not be able to change the temperature parameter (Fig. 6).

In a third MVP, the parameter modification functionality was already implemented, and the user could interact with the internal climate of the greenhouse (Fig. 7).

The next deliverables of the project were based on the user's experience, such as design and response performance.

The modeling using the Lean Inception methodology requires that the visualization is more focused on the business with less focus on the functionalities since the modeling already tells what should be done, the development team focuses on fulfilling the functionality so that it works.

In this type of modeling visualization, the client's understanding is better considering that the client's concern is focused on the functioning and not how it is created.

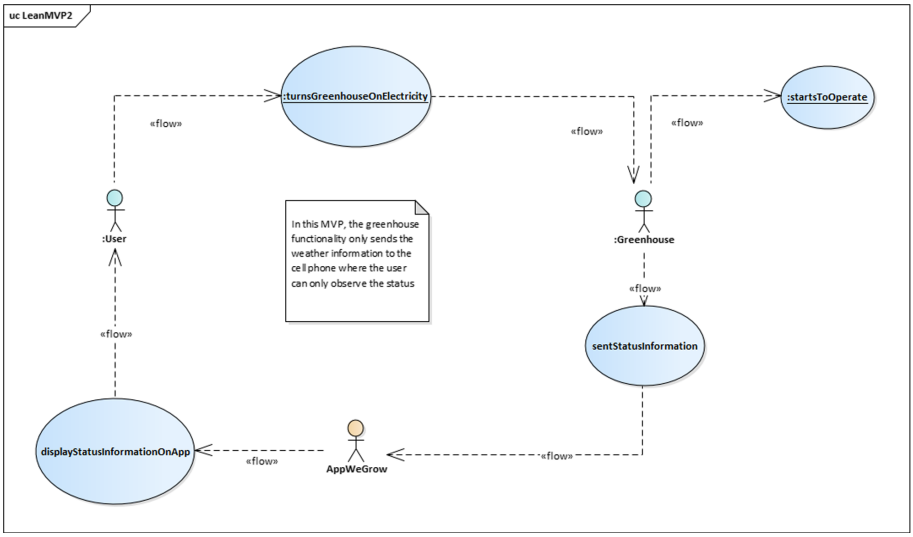

Fig. 6. Second MVP UseCase delivery

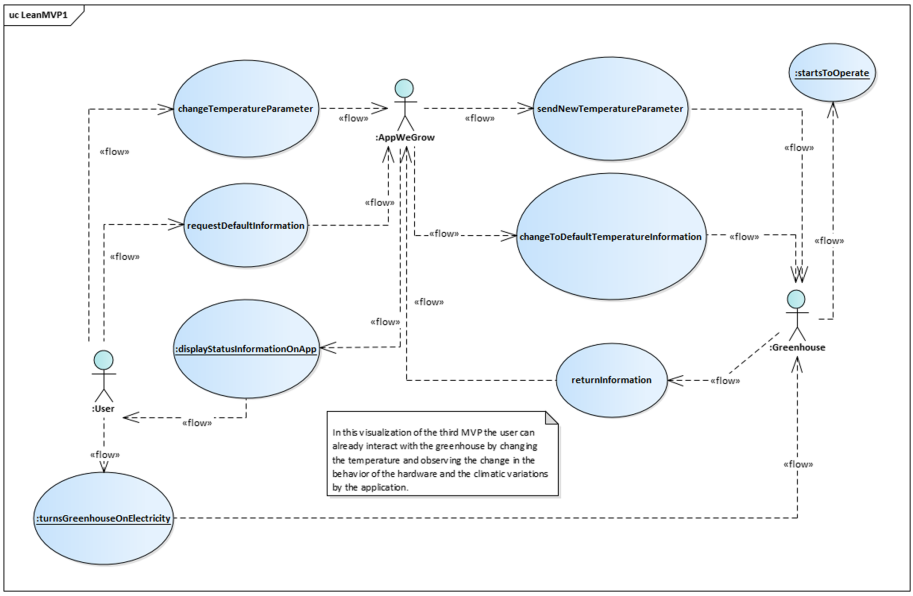

Fig. 7. Third MVP UseCase

5 Discussion

Depending on the methodology applied in a software, the project requires modeling with other levels of understanding and abstraction.

In the paper two agile methodologies that require different modeling methods and design dynamics were approached, where both have similar results but serve for different types of projects, the diagrams made in the same project but in different methodologies were highlighted in the article for a better visual understanding.

As discussed the focus of both methodologies are agile deliveries that bring a quality result to the client - lean inception with fractional deliveries of a MVP and Scrum with deliveries divided by sprints. The difference is the time the user will be able manipulate the application. When using lean inception the first delivery is already viable for the user. When using Scrum delivery will not necessarily be usual.

6 Conclusion

It is concluded that different development methodologies require different levels of abstraction in modeling and architecture, no matter how close they are, and their system requirements are the same, each method requires a different assessment.

Scrum for example, covers more with regard to the content of the system (whether functional or non-functional), the behavior of the application and how it will measure in the use of the final product.

Lean Inception, on the other hand, requires a higher level of abstraction, taking better care of the stakeholders' difficulties, evaluating all the processes involved so that there is satisfaction from both parties (development team and business owners).

Acknowledgments. This work was supported by FCT - Foundation for Science and Technology within the scope of the R&D units project: UIDB/00319/2020. This research is sponsored in collaboration between the Incentive System to Portugal for Technological Research and Development and Universidade Paulista UNIP - Research Group on Software Engineering in Brazil.

References

1. Pressman, R., Maxim, B.: Software Engineering: A practitioner's approach, 8th edn. McGraw-Hill Education, London (2015)
2. Bruno Adam Osiek. CHAOS Report (2011). Métodos Ágeis Aumentam Taxa de Sucesso de Projetos https://blog.myscrumhalf.com/metodos-ageis-impactam-positivamente-no-sucesso-de-projetos/. Accessed 26 Feb 2020
3. Mountain Goat Software. Scrum (2019). https://www.mountaingoatsoftware.com/agile/scrum . Accessed 5 Feb 2020
4. Caroli, P.: Lean Inception - How to Align People And Build The Right Product. Editora Caroli, Sao Paulo (2018)
5. OMS. Anxiety (2017). https://www.who.int/eportuguese/countries/bra/pt/. Accessed 20 Jan 2020
6. Palhares, I.: Brasil Tem a Maior Taxa de Transtorno de Ansiedade do Mundo, diz OMS (2017). https://saude.estadao.com.br/noticias/geral,brasil-tem-maior-taxa-de-transtorno-de-ansiedade-do-mundo-diz-oms,70001677247. Accessed 25 Feb 2020