FACULTY OF SCIENCE AND TECHNOLOGY

MASTER THESIS

Study programme / specialisation:
Industrial economics / Innovation and technology management

The spring semester, 2022

Open / ~~Confidential~~

Authors:
Simon Degen
Frøydis Jørgensen

Simon Degen Frøydis Jørgensen
(signature author)

Course coordinator: Frank Asche

Supervisor(s): Sindre Lorentzen

Thesis title:
A Case Study: Issues when implementing Lean Startup, Design Thinking, and Scrum.

Credits (ECTS): 30

Keywords: Lean Startup, Design Thinking, Scrum, Double Diamond, Agile, Kanban, Scrumban, Software Development, Startup

Pages: 75

+ appendix: 4

Stavanger, June 14th/2022
date/year

A Case Study:

Issues when implementing Lean Startup, Design Thinking, and Scrum

Simon Degen and Frøydis Jørgensen

June 2022

Universitetet
i Stavanger

Department of Safety, Economics and Planning
Faculty of Science and Technology
University of Stavanger

Abstract

Motivation: Lean Startup, Design Thinking, and Scrum are three methodologies with increased use in software development projects. Although they are well-known methodologies, there are risks of encountering problems during implementation. To learn more about the type of problems and how they may be solved, we conducted a case study examining issues related to the implementation.
Method: Our case organization is a software startup. The qualitative data about the organization were collected through eight semi-structured interviews with people representing all the company departments. The analysis of the collected data has been done according to the five-phased cycle described in Yin (2011).
Findings: Time pressure was caused by poor project definitions, unrealistic time estimates, and insufficient knowledge about the methodologies. The scope of the Minimum Viable Product (MVP) was too large, so important concepts in Lean Startup were not executed. Using the Double diamond model approach, Design Thinking was performed mainly focusing on solving problems instead of exploring problems. Later on, the entire Double diamond model lost priority due to time pressure. In addition, Scrums' lack of guidance on backlog management resulted, in this case, looking similar to a waterfall method with Scrum ceremonies.
Conclusion: With a better definition of the MVP, the Lean Startup Build-Measure-Learn feedback loop may have been achievable. The company's structure prevented the implementation of some Lean Startup principles and structural attributes, which resulted in a lack of startup mentality. The Double diamond model could have strengthened the vision and potentially prevented the vision's derailment by focusing more on exploring the problem instead of the solution. Testing and verifying the design seems vulnerable to neglection under time pressure. The issues with Scrum backlog management, estimation, and time pressure led to problems finishing sprints and added pressure on the team.

Acknowledgements

First and foremost, we would like to thank the organization that has given us this exciting opportunity and provided guidance and support along the way. A special thanks for allowing us to be a part of your daily environment, which has given us easy access to information and many pleasant working days. It has solely been a positive experience working with you. We would also like to thank our supervisor, associate professor Sindre Lorentzen at UiS, for their help with writing this thesis.

Contents

Abstract **i**

Acknowledgements **ii**

1 Introduction **1**

1.1 Objective 1

1.2 Motivation 2

1.3 Novelty 2

1.4 Approach 3

1.5 Outline 4

2 Background **5**

2.1 About the company 5

2.2 The market 6

2.3 Team structure and organization 7

3 Theory **8**

3.1 Software development 8

3.1.1 Frontend development 9

3.1.2 Backend development 10

3.2 Lean Startup 11

3.2.1 The method 11

3.2.2 The process 13

3.2.3 Critical success factors for Lean Startup 15

3.2.4 Disrupt innovation by Lean Startup 16

3.3 Design Thinking 17

3.3.1 The Double Diamond model 18

3.4 Agile 20
3.4.1 Scrum 21
3.4.2 Kanban 24

4 Literature Review 26
4.1 Critique of Lean Startup 26
4.2 Critique of Design Thinking 27
4.3 Critique and known challenges in Scrum 28
4.3.1 Backlog management 28
4.3.2 Estimation 29
4.3.3 Communication in the COVID-19 pandemic 29
4.4 InnoDev 30
4.5 Joint Venture Startups 31

5 Method 33
5.1 Data collection 33
5.2 Data processing 34
5.3 Quality of the data 36
5.3.1 Avoiding researcher bias 37
5.4 Ethical considerations and data management 37
5.5 Citing the interviews 37

6 Findings 39
6.1 The structural attributes 39
6.2 Management and crucial decisions 40
6.3 Development of the MVP 42
6.4 Design Thinking implementation 43
6.5 Testing and feedback 44
6.6 Implementing Scrum 45
6.7 Backlog management 46
6.8 Deadlines and estimation 46

7 Discussion 48
7.1 Key Findings 48
7.1.1 Lean Startup 48
7.1.2 Design Thinking 49

7.1.3 Scrum 49
7.2 Partly implementation of Lean Startup 50
7.3 Deadlines and collaboration affecting the Double Diamond model 52
7.4 Reviewing Scrum implementation issues 53
7.5 The importance of Scrum knowledge 54
7.6 Backlog management 54
7.7 Estimation and agility of Scrum 55
7.8 COVID-19's affect on Scrum 56
7.9 Joint venture affecting the implementations 57
7.10 Iterating methods 58

8 Conclusions **59**
8.1 RQ1: What were the issues for ORG when implementing Lean Startup? 60
8.2 RQ2: What were the issues for ORG when implementing Design Thinking? 60
8.3 RQ3: What were the issues for ORG when implementing Scrum? 61
8.4 Further recommendations 61

A Interview Guide **69**
A.1 Introduction 69
A.2 Main section 69
A.3 Summary/Conclusion 70

B Intervjuguide **71**
B.1 Introduksjon 71
B.2 Hoveddel 71
B.3 Oppsummering/Avslutning 72

List of Figures

2.1 The organization structure. 7

3.1 Frontend architecture in Piral (Piral, 2022). 10
3.2 A comparison of a classical backend to a micro service backend (Yang et al., 2019). 10
3.3 The build-measure-learn feedback loop (Ries, 2011, p.47). 12
3.4 The process of Lean Startup (Eisenmann et al., 2012). 14
3.5 The Double Diamond model from the British Design Council (Design Council, 2019b). 18
3.6 Scrum VS waterfall mode (Unyscape, 2019). 20
3.7 The Scrum process visualized(Schwaber & Sutherland, 2011, p.18). 21
3.8 Scrum in a nutshell (Kniberg & Skarin, 2010, p.3). 23
3.9 The Kanban board (Kniberg & Skarin, 2010, p.15). 24

4.1 The three-phase InnoDev process (Dobrigkeit, de Paula, et al., 2017, p.326). 31

5.1 The five-phased cycle (Yin, 2011, p.177). 35

7.1 Illustration of what the process in the start-up phase may have looked like. This is not an exact representation. 56

8.1 A summary of the conclusion. 59

Chapter 1

Introduction

Work methodologies such as Lean Startup, Design Thinking, and Scrum are rising in popularity within software development. However, when implementing them, there is a risk of encountering issues. With the increase in usage, uncovering issues so that the methodologies can be implemented efficiently is important. Scrum specifically increased its popularity and usage in organizations using agile development methodologies from 40% in 2007 to 70% in 2018 (Hoda et al., 2018). This case study aims to discover issues encountered when implementing Lean Startup, Design Thinking, and Scrum. It is done in collaboration with a software startup company. The company was founded in 2020 as a joint venture by a software company and a large telecommunication (telecom) company and is focusing on a growing eSports market.

This case study is written anonymously to take into consideration the startup company, as we do not want to affect future investors. We will refer to the company with the alias ORG throughout the thesis.

1.1 Objective

Starting this case study, we knew that the initial phase of ORG, like many other companies, did not go optimally. Before beginning this study, we were told that Lean Startup, Design Thinking, and Scrum methodologies were applied. By collecting the story of ORG from several people inside the company, we wanted to see how applying these methodologies, which are intended to be beneficial, may create new problems or be harder to implement than initially thought.

This research aims to look at the essential elements that must be present and potential issues when implementing these methodologies. To answer this, three research questions have been defined.

- RQ1: What were the issues for ORG when implementing Lean Startup?
- RQ2: What were the issues for ORG when implementing Design Thinking?
- RQ3: What were the issues for ORG when implementing Scrum?

ORG is a company in constant change. Therefore, we defined a time period to examine. We will be looking at a time period of ORG's first six months.

1.2 Motivation

As developers, the concepts of Lean, Design Thinking, and Agile is likely to impact our work. In addition, this is not only relevant for developers, but everyone involved in Information technology (IT) projects as these are methodologies with increasing use. Although these methodologies are based on making processes simpler and more flexible so that customers and users get the best experience, there are several well-known problems you may encounter along the way (Akif & Majeed, 2012; Iskander, 2018; Lizarelli et al., 2021). This case study is conducted to learn more about the issues one may encounter when implementing Lean Startup, Design Thinking, and Scrum.

1.3 Novelty

Multiple software development work methodologies are rising in popularity, entailing companies to use them individually, combined, or modify them to fit their case perfectly. Agile methodologies have been a massive research topic, with recommendations on how to use Scrum and other Agile methodologies most efficiently. Still, implementation is rarely entirely optimized, and with different methodologies combined or an unusual company structure, complexity increases. Signoretti et al. (2020) suggests further research on the combined approach to compare its strengths and weaknesses to other agile methodologies. This case study will not answer the exact suggestions. Still, it can be used as an example where Lean Startup, Design Thinking, and Scrum are used within a company without coordinating the implementations, contrary to hybrid models, such as InnoDev. In addition, Zorzetti et al. (2022) requests experiments to compare the combined approach such as this case represents to InnoDev and other individual approaches.

York and York (2019) suggests further research on Lean Startup's gaps in implementation and how it can be used with other methodologies. In addition, Canedo et al. (2020) wants to investigate the main challenges when using Design Thinking in software development teams, where this case study can assist as an example case study.

Overall, there is still a visible need for further research and real-context cases on implementing Lean Startup, Design Thinking, and Scrum. As far as our knowledge goes, a similar case study has not been done before.

1.4 Approach

Our main approach for data collection was primarily semi-structured qualitative interviews based on works by Robert K.Yin. We conducted a total of 8 interviews with employees of ORG. The average duration of an interview varied from 40 minutes to 1 hour and 20 minutes. In addition to this, we have continuously had access to the employees for follow-up conversations. Once the data was collected, we created a database of all the interviews. Using a five-phase system cycle (Yin, 2011), we analyzed and sorted the data such that we could present it in the findings chapter. We have carefully considered the quality of the data and taken. As one of the students has worked closely with ORG earlier, we have taken several measures to avoid researcher bias. One of the steps taken is to remove this person from the interviews.

1.5 Outline

In this thesis, we have used the following layout:

- **Chapter 2** presents and discusses background information about ORG and the market.
- **Chapter 3** presents the theory associated with this thesis and our research questions.
- **Chapter 4** presents the literature we have used as a foundation for our arguments and discussion.
- **Chapter 5** presents the methods we have used in this case study to collect and analyze data.
- **Chapter 6** presents the findings and results we have found in our research.
- **Chapter 7** discusses the findings and results in terms of our research questions.
- **Chapter 8** presents a conclusion and summary of our findings and discussion.

Chapter 2

Background

2.1 About the company

In this case study, the company ORG we refer to is a Norwegian software company founded in 2020. It is a joint venture by one large telecom company, *Partner A* and one IT-consulting company, *Partner B*. As a response to the increasing popularity of eSports in the world, ORG decided to capitalize on the increasing amount of casual gamers. They aim to provide a platform that makes it easy to organize and participate in eSport. To achieve this, they will tailor their design toward a more casual gamer and make the user experience better for individuals who have not yet devoted many hours to these kinds of activities. To further increase their knowledge and chance of success, they partnered with a sports event organizer, *Partner C*. These owners and partnerships provide ORG with a wide range of expertise, from streaming, distribution, software development, and hosting large sports events.

A differentiating factor of ORG is the customer segment they are targeting. As several other contenders in the market already have platforms and arenas created to accommodate pro gamers, ORG has chosen to target casual gamers that typically do not participate in eSport events and instead simply play with friends to have fun. However, they also wish to draw in experienced gamers through a high standard in terms of customization and quality.

Development started in January 2021, while concept development began a few months prior. They planned on using key strategies such as Lean Startup, Design Thinking, and Scrum. When development began in January, the team consisted of two designers, five software developers, and one project leader.

2.2 The market

To understand the opportunity and market ORG wants to succeed in, we must first understand the aim of ORG. According to the CEO of ORG, they want to make it easier for everyone to enjoy the thrill of eSport without the inconvenience of organizing a game. As mentioned earlier, they are targeting both casual and more experienced gamers. They want to reach these individuals through schools, sports clubs, and firms. They have targeted these groups because they believe this will increase their userbase and rate of adoption as they are targeting groups instead of individuals.

According to the Ministry of Culture (2019) more than half of the Norwegian population play video games in some form. This fact shows that there are users in the market. However, we must recognize that there are other established competitors in the market. Amongst some are Gamer.no and Esportal. They are both platforms tailored toward gaming experiences. However, an advantage for ORG is that Esportal has described itself as “a web-based competitive platform for the major eSport titles, currently focusing on CS:GO” (Esportal, 2022). As previously mentioned, ORG is not aiming for competitive users but rather casual gamers.

The eSports industry is growing. Newzoo is one of the world’s most trusted sources for games and eSports analytics and market research (Knoema, 2022; Newzoo, 2022). They have created a forecast that claims the growth in the global games market is expected to continue in its current fashion in the following years and estimate that there will be 2.8 billion gamers globally in 2021 (Newzoo, 2021). In addition to this, they expect a compound annual growth rate of +7.2% from 2019 such that revenues exceed $200 billion in 2023 (Wijman, 2021). Even though this proves that there is a large and growing global market, it is worth noting that it would not be easy for a small firm to stand out as there are many hundreds of international competitors (Tracxn, 2021).

2.3 Team structure and organization

As mentioned earlier, ORG is a joint venture between a large telecom firm and a software firm. These two companies manage ORG through the board of directors, as we can see in fig 2.1. ORG has tried to capitalize on the strengths of the respective firms by hiring developers from the software firm and management from the telecom firm, as these were available for a reduced price. The structure of the firm is shown below in Figure 2.1.

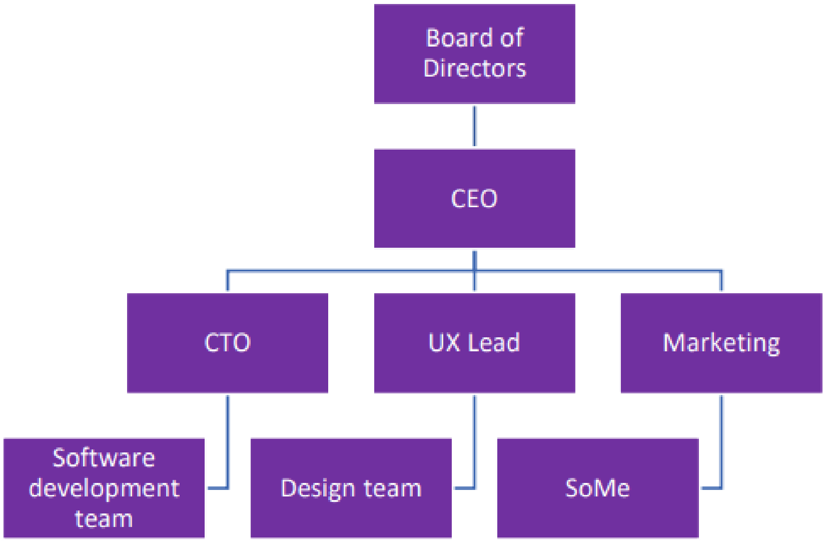


Figure 2.1: The organization structure.

In the period we are looking into, the amount of employees has fluctuated. There were a total of 5 software developers at the beginning, which shifted to 3 and finally up to 5 again. On the other hand, the amount of designers has been constant at two designers. In addition, a person was later on hired to fill the social media (SoMe) position.

Chapter 3

Theory

This chapter will review theories that are relevant to our research questions. First, there will be a brief explanation of software development, focusing on relevant concepts. Then we will go through the working methodologies used by ORG, Lean Startup, Design Thinking, and some agile methodologies with a focus on Scrum. This chapter aims to provide the reader with a sufficient understanding of the work methodologies.

3.1 Software development

> “Software development refers to a set of computer science activities dedicated to the process of creating, designing, deploying and supporting software” - (IBM, 2022)

The process of developing software is usually done by software developers. Software developers are responsible for writing code as well as driving a project through a software development life-cycle. This entails working in teams to meet the requirements the customer sets and transforming these requirements into features. In addition, they can also be responsible for testing and maintenance of the product and code(IBM, 2022). When referring to software developers, it may be split further into two different roles frontend developers and backend developers. There is also a third variant called full-stack developers. However, in this project, this role was not present and will therefore not be explained or discussed further.

3.1.1 Frontend development

Frontend development is the practice of producing a website or web application that a user can interact with; therefore, another name for it is client-side development. These applications and websites are usually created with technologies and coding languages known as CSS, HTML, and Javascript. A developer can use one or several of many technologies to aid in the production of these websites, some of these tools are called frameworks, and examples of these include React, Angular, and Vue (Frontendmasters, 2022). In addition to tools, we can explore different architectural choices; one of these is micro frontends.

Micro frontend architecture

According to Yang et al. (2019), the idea behind micro frontends is to treat a web application as a combination of features owned by different teams. Where each team has an independent function and business that they focus on. This is based on a similar concept that has been around for a while called micro-services, a service-oriented architecture style that builds applications as a collection of loosely coupled services. This is an architecture that has been used in the backend for a while and can be seen in Figure 3.2. So when we implement a micro frontend architecture, several core ideas should be reflected. Each team should be able to complete features and tasks in their section independently from other teams; this entails that the team's code should be isolated. This means that it does not affect the other team's code. By doing this, we can avoid sharing run-time and global variables.

In general, micro frontends are beneficial in projects that depend on a scalable architecture. As Fowler (2019) states, it should be considered if technical and organizational maturity is present, or else such an approach might lead to more chaos.

An example of a micro frontend is Piral. It works by having shared libraries and patterns for all teams. These live in a component called the app shell. The app shell is the place where all the teams come together. In Piral, each team works on a module which can be referred to as a pilet. These pilets are then connected and displayed in the appshell, as shown in Figure 3.1. (Piral, 2022)

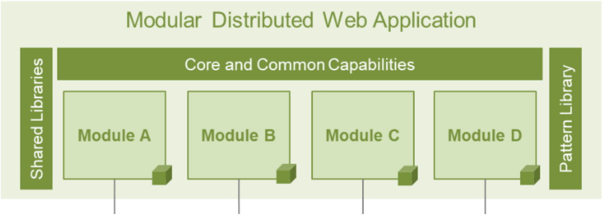

Modular Distributed Web Application
Core and Common Capabilities
Shared Libraries
Module A
Module B
Module C
Module D
Pattern Library

Figure 3.1: Frontend architecture in Piral (Piral, 2022).

3.1.2 Backend development

As frontend development can be referred to as client-side development, and backend, on the other hand, can be referred to as server-side development. This means that backend development handles everything that happens on the servers and databases. So when a website requires data or data processing, this is typically done on a server that backend developers create and manage. Similar to frontend development, there are many powerful tools to help the developer write code, such as .Net and Node.js, as well as different architectural possibilities. One of the most popular architectures is the micro-service approach, as seen in Figure 3.2 (Shiotsu, 2021).

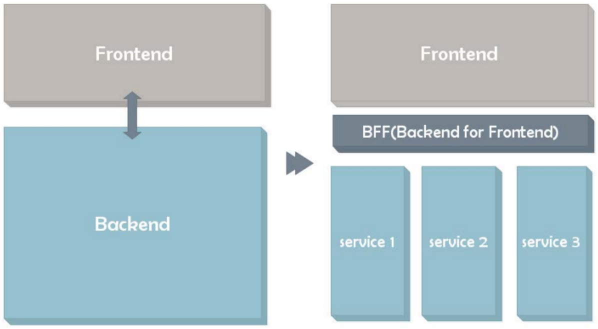


Figure 3.2: A comparison of a classical backend to a micro service backend (Yang et al., 2019).

3.2 Lean Startup

Lean principles can be traced all the way back to the early seventies and were developed by Toyota in Japan. The concept was to reduce waste in processes, which led to more efficient production processes. (Müller & Thoring, 2012). The book "The Machine That Changed the World" (Womack et al., 1990) followed by "Lean thinking—banish waste and create wealth in your corporation" (Womack & Jones, 1996) shifted organization's focus on creating a more customer-oriented product while reducing waste. To be competitive today in manufacturing, some sort of lean process is considered necessary. Still, it can also be crucial for service operations as it is about ongoing learning and continuous improvement at its core (Liker & Morgan, 2006).

Most startups fail; in fact, more than 9 out of 10 startup companies fail (Cerdeira & Kotashev, 2021). The Lean Startup method is developed by Eric Ries (2011), with a mission to help entrepreneurs succeed and improve the success rate of startups. It is based on initial lean principles; innovation is the most effective when users demand it. That is to say, you create waste when the product is not needed by the users, which is contradictory to lean principles (Liker & Morgan, 2006). The definition of a startup varies, but according to Ries, "A startup is a human institution designed to create a new product or service under conditions of extreme uncertainty." (Ries, 2011, p.22). In other words, a startup is an experiment.

3.2.1 The method

The Lean Startup method is a scientific approach to increasing the speed of a startup. The method helps businesses drive a startup, how to steer, when to preserve, and how to grow a business with maximum acceleration (Ries, 2011). It consists of five principles;

Entrepreneurs are everywhere

Entrepreneurs are not just to find in small newly created companies, but in any size company in any sector or industry. According to Ries (2011), the concept of entrepreneurship includes anyone who works within a startup. Ries's definition of a startup is comprehensive, and with tons of existing opportunities, entrepreneurs are everywhere.

Entrepreneurship is management

As stated in Ries's definition, a startup is an institution. Therefore, it requires management to customize explicitly to the context of dealing with extreme uncertainty (Ries, 2011). Being flexible and learning-oriented is essential for a startup's success.

Validated learning

Lean startups exist to learn how to build a sustainable business, and that learning can be validated scientifically by running frequent experiments. This means building your business on results and relevant data and testing each element in their vision (Ries, 2011).

Build-measure-learn

The build-measure-learn feedback loop is considered the key element of Lean Startup and should be done continuously. Testing your hypothesis and getting feedback on your product is essential for creating a product customers want. Using a prototype or an MVP of the product, customer response should be measured, and then learn whether to pivot or persevere. The idea is to minimize the total time by accelerating the feedback loop, as shown in Figure 3.3 (Ries, 2011).

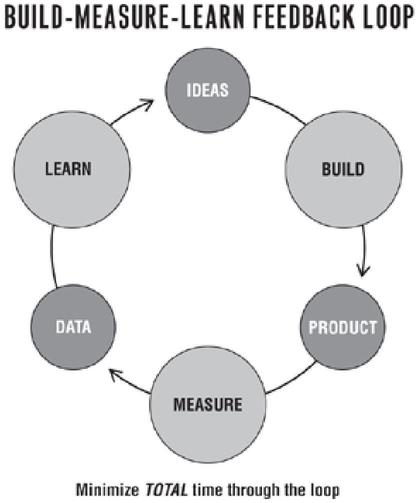

BUILD-MEASURE-LEARN FEEDBACK LOOP
IDEAS
BUILD
PRODUCT
MEASURE
DATA
LEARN
Minimize *TOTAL* time through the loop

Figure 3.3: The build-measure-learn feedback loop (Ries, 2011, p.47).

Innovation accounting

Focusing on how to measure, set milestones, and prioritize work based on data is needed to improve the outcome and hold innovators accountable. In other words, it is about taking the measuring and learning phases from the feedback loop and developing insights into engagement, product-market fit, and scalability (Ries, 2011).

3.2.2 The process

Lean startup is a method and not a process. However, by using the method, a process similar to as shown in Figure 3.4 is likely to ensue. Lean startup is based on hypothesis-driven entrepreneurship but differs from previous hypothesis-driven processes such as “Build-It-And-They-Will-Come” or “Just Do It!” approaches. Where the “Build-It-And-They-Will-Come” approach is having a strong vision of the product and initially having a direction, feedback will be bypassed. On the other hand, the “Just do it!” approach takes feedback carefully into consideration but lacks structure, which can lead to costly and time-consuming detours. Lean startup aims to test a set of business model hypotheses by having the structure elements present while having a consumer focus to streamline the process (Eisenmann et al., 2012).

Before generating the hypothesis, every entrepreneur needs a vision for the problem the startup will address and create a solution for. This step is often called ideation and includes brainstorming concepts and theories that could potentially be implemented. Based on this vision, a business model hypothesis is created where topics such as customer segments, market plan, cash flow formula, and technology and operation management should be addressed. This should be formulated; hence it can be falsified (Eisenmann et al., 2012).

The next step is to specify the minimum viable product (MVP) based on the hypothesis. The MVP is an early, simplified version of the new product or service. It should be the simplest version possible but contain enough to generate informative feedback about the nature of the product. The goal is to confirm or falsify at an early stage the hypothesis of why the new product or service will succeed (Ries, 2011). In other words, the smallest set of activities needed to disprove a hypothesis (Eisenmann et al., 2012).

After testing the MVP and getting feedback, the team must decide whether

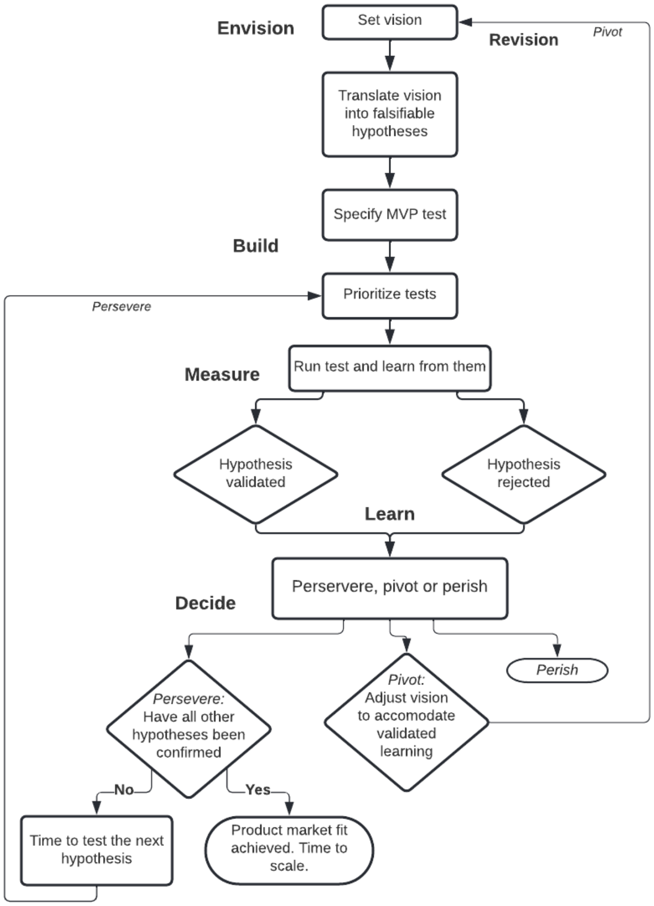

Envision
Set vision
Revision
Pivot
Translate vision into falsifiable hypotheses
Specify MVP test
Build
Prioritize tests
Persevere
Run test and learn from them
Measure
Hypothesis validated
Hypothesis rejected
Learn
Perservere, pivot or perish
Decide
Perish
Persevere: Have all other hypotheses been confirmed
Pivot: Adjust vision to accomodate validated learning
No
Yes
Time to test the next hypothesis
Product market fit achieved. Time to scale.

Figure 3.4: The process of Lean Startup (Eisenmann et al., 2012).

to persevere, pivot, or perish. If the MVP validates the hypothesis and no feedback prompts a significant change in direction, persevering is the right thing to do. That means continually improving the product or service with minor changes and updates according to the feedback and progression towards the goal, without changing the vision. If all hypotheses are validated, the team can start preparing to scale (Eisenmann et al., 2012; Ries, 2011). If the MVP falsifies the hypothesis or the hypothesis is verified, but the feedback remarks that better opportunities lie elsewhere, the entrepreneurs may choose to pivot (Eisenmann et al., 2012). Pivoting is changing the fundamental hypothesis in a structural modification of the product and strategy while keeping the original vision (Ries, 2011). Pivoting can be done in multiple dimensions of a business model. For example, the business may change the customer value proposition by either changing the feature set, the customer set, or both. Pivoting may also include changes in market plans, cash flow formula, or technology and management strategy. If testing the MVP rejects the hypothesis and there is no reasonable pivoting, the business should be shut down (Eisenmann et al., 2012).

3.2.3 Critical success factors for Lean Startup

Critical success factors are elements that an organization needs to get right to ensure success (Boynton & Zmud, 1984). As mentioned in section 3.2, Lean Startup is designed to help startups deal with risks and increase development speed while keeping a customer-oriented focus. This section will discuss critical success factors regarding the methodology and suggested process.

MVP and customer feedback

Defining and designing MVPs and testing priorities is challenging as multiple variables need to be decided carefully. Some challenging variables are choosing the correct consumer group, defining statistical parameters of group size, and typing and obtaining information. Customer feedback should be from the relevant cluster of customers, and the goal is to understand their needs and requirements. In addition, preserving a high level of customer involvement and engagement is likely a challenge. While testing and verifying the product, maintaining its credibility and business identity is essential as they risk presenting a rough prototype with significant flaws or unwanted features. There is also a risk of ending up with multiple MVPs, which can lead to increased cost and a slower process (Lizarelli et al., 2021). Finding a balance of making an MVP that satisfies consumers but is simplified enough to enter and capture the market quickly is crucial for success.

Pivoting

Figuring out when to pivot can be difficult as it is typically only considered when the hypothesis is falsified but can also be based on feedback even when the hypothesis is verified. Realizing pivoting may be the best option for the company can lead to a feeling of failure and demotivate the team (Lizarelli et al., 2021). Having a personal stake in the outcome and keeping the motivational factors high, as mentioned in section 3.2.4, is important to maintain the team's drive. The team should also be aware that pivoting is not equivalent to failing, as it is just a restructuring and learning opportunity that can increase the chance of success. Another challenging but crucial task is to figure out how to pivot. What changes need to be done, and how should the changes be executed. Having a systematic way to pivot, which enhances the learning opportunity, is a critical success factor (Lizarelli et al., 2021).

3.2.4 Disrupt innovation by Lean Startup

The theory of disruptive innovation, developed by Clayton M. Christensen, refers to a process of converting an expensive or complex product or service into a more affordable, simpler product that is accessible to a broader population. It requires an innovative business model, enabling technology, and a coherent value network (Dan & Chieh, 2008). According to Ries (2011), three structural attributes are needed for startup teams to nurture disruptive innovation; (i) scarce but secure resources, (ii) independent authority to develop their business, and (iii) a personal stake in the outcome. Getting the structural attributes right will not necessarily lead to success, but getting them wrong can almost certainly lead to failure (Ries, 2011), meaning these structural attributes are the structural critical success factors.

(i) Scarce but secure resources

A startup is more exposed to budgetary changes with less room for error. In larger, well-established organizations, the project will try to acquire as large a budget as possible. They usually have some padding in case of a crisis somewhere else in the organization. This differs from startups, where both having too much and too little budget is harmful. Startups are easier to manage and need less capital. Having too much might acquire the startup an expensive process without increasing productivity and value, as they initially have the advantage of being cheaper to manage. They are also sensitive to budgetary changes, and losing some of their budgets could be fatal. Therefore, they need less capital overall, but the capital must be secure from tampering (Ries, 2011).

(ii) Independent development authority

Parent organizations usually have strong opinions regarding the development of the new product, but the startup team should be autonomous. A cross-functional team with open communication and full-time representation in every functional department is strongly recommended, but the team should be kept to a minimum (Lizarelli et al., 2021; Ries, 2011). The team should independently do the build-measure-learn feedback loop without parent organizations, as they will slow down the process. Allowing startup teams to be independent and autonomous will increase speed and creativity, but as the level of autonomy is high, avoiding raising the parent organization's fears is essential (Ries, 2011).

(iii) A personal stake in the outcome

One recommended method to avoid fears in the parent organization is to ensure that the entrepreneurs have a personal stake in the outcome. Financial objectives such as stock options or other equity ownership systems and reward systems are incentives to increase long-term performance. The rewards system does not necessarily need to be financial as long as it increases the personal stake in the outcome (Ries, 2011) and increases motivation. Increasing the intrinsic motivation is shown to increase the team's engagement and is a more long-term approach than extrinsic motivation (Mone et al., 2011). Building a good company culture and great workplace is one method to increase personal stake in the outcome without financial incentives (McGregor & Doshi, 2015).

3.3 Design Thinking

According to Tschimmel (2012), Design Thinking can be understood as a complex thinking process of conceiving new realities, expressing the introduction of design culture and its methods into fields such as business innovation. Since Design Thinking can be understood as a thinking process and not a set of confining "do's and don'ts", there has been space for several different models and ways to implement Design Thinking to emerge. Amongst these models, the most known models are the 3 I model, the HCD model, the Double Diamond model, and the Design Thinking model of Hasso-Plattners institute (Tschimmel, 2012). The model relevant in this case is the Double Diamond model.

3.3.1 The Double Diamond model

The Double Diamond design model was officially invented by the British Design Council in 2005. It was produced to give a simple graphical way of describing the design process (Design Council, 2007). They created this model through extensive research with several large brands. They uncovered that many firms were doing the same processes but with different names and structures. The resulting model proposed by the British Design Council can be divided into four distinct sections. Discover, define, develop and deliver. These stages contain both divergent and convergent sections, which in practice refers to exploring the issues more widely or narrowly. Contrary to the arrows often shown in the visual representation of the Double Diamond, the process is not linear. The British Design Council has stated that when new information or challenges arise, it can be beneficial to return to the start of the process (Design Council, 2019b).

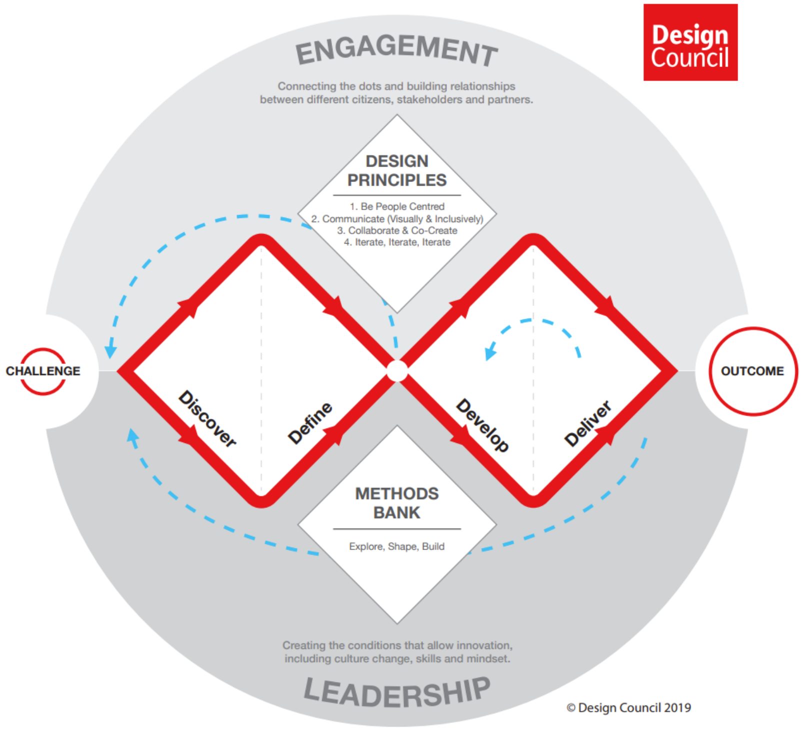

Design Council
ENGAGEMENT
Connecting the dots and building relationships between different citizens, stakeholders and partners.
DESIGN PRINCIPLES
1. Be People Centred
2. Communicate (Visually & Inclusively)
3. Collaborate & Co-Create
4. Iterate, Iterate, Iterate
CHALLENGE
Discover
Define
Develop
Deliver
OUTCOME
METHODS BANK
Explore, Shape, Build
Creating the conditions that allow innovation, including culture change, skills and mindset.
LEADERSHIP
© Design Council 2019

Figure 3.5: The Double Diamond model from the British Design Council (Design Council, 2019b).

Discover

The first step in the model is discover. This stage is supposed to help users, and people understand the issue or problem. This is often done by talking to the people that have experienced the given issue or problem. Other typical activities that can be done in this stage are market research and user testing (Design Council, 2019b).

Define

This stage focuses on working through the data gathered in the discover phase. If the previous stage can be considered a diverging stage, this stage can be viewed as convergent because the work being done is to narrow down the scope. Once the gathered data and insights are processed, the challenge can be approached and defined in a new way. According to the British Design Council, the result of this stage is to create a design brief that clearly defines the challenge based on the insights (Design Council, 2019a).

Develop

In this stage, the key is to develop, iterate, and test possible solutions. As can be seen in Figure 3.5, this stage is a convergent stage, meaning that solutions can be explored more widely. According to the British Design Council, there are several key activities including the following: Visual management, Development methods, and Testing (Design Council, 2007).

Deliver

Finally, the last stage of the process is the deliver stage. As the name suggests, it refers to the delivery of the product or service. There are most likely several possible solutions from the previous stage. However, in this stage, a single solution needs to be selected. It is typical to work on refining and finalizing the solution as well as launching the product in the market. In addition to this, the launch and finalizing of the product also signalize the start of the feedback loop. Typical key activities in this stage are; Final testing, approval and launch, creation of targets, evaluation, and feedback loops (Design Council, 2007).

3.4 Agile

To be agile means being able to “Deliver quickly. Change quickly. Change often” (Highsmith, 2004, p.15). There are many different agile techniques that vary in emphasis and routines, but there are some common characteristics. These include the process of iterative development, the reduction of resource-intensive intermediate artifacts, and finally, a focus on communication and interactions. Having all of these characteristics should add maneuverability and therefore allow an agile project to identify and respond to changes faster than traditional approaches (Cohen et al., 2004).

At the beginning of 2001, seventeen Agile proponents came together to discuss new and emerging software development methodologies. The meeting resulted in the creation of the Agile Manifesto (Fowler, Highsmith, et al., 2001). The growth of agile development methodologies originated as a reaction to traditional ways of software development, which were deemed to be too slow-paced for an increasingly fast-paced world in the 1990s (Cohen et al., 2004). Another well-known agile method is the waterfall model, a sequential development model. In this model, each task and all requirements should be clear before moving on to the next task. Contrary to Scrum, which will be explained below, there should be no overlapping of phases, and everything should be done in a linear way (Balaji & Murugaiyan, 2012). In Figure 3.6, the difference between linear (waterfall) and iterative (Scrum) methods are shown.

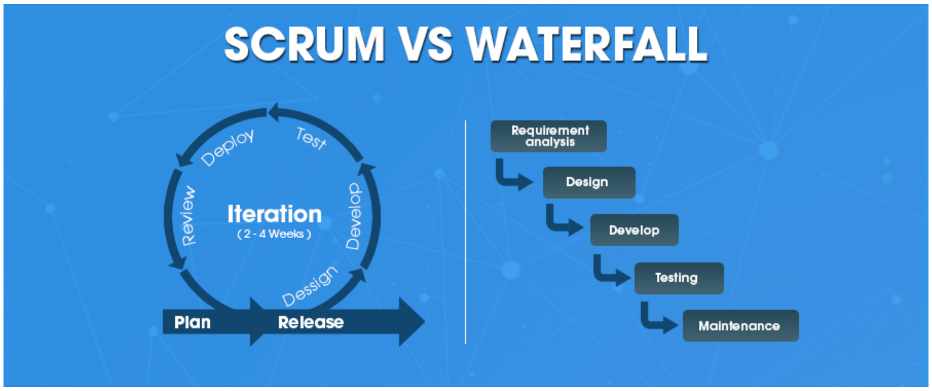


Figure 3.6: Scrum VS waterfall mode (Unyscape, 2019).

3.4.1 Scrum

This section is based on the Scrum guide from Schwaber and Sutherland (2011). Scrum can be defined as a lightweight framework that helps people, teams, and organizations generate value through adaptive solutions for complex problems. To successfully implement Scrum, one must first have a fundamental understanding of the roles and ceremonies that occur in the Scrum development process. The Scrum team typically consists of few members, where each team has a Scrum master, product owner, and the rest of the team. The roles of the different members are as follows:

- Scrum master: The team member that takes this role is responsible for making sure the team follows the Scrum guidelines. The Scrum Master is also responsible for the team's efficiency and can be viewed as a servant-leader.
- Product owner: The primary responsibility of the product owner is to represent the customers and the stakeholders. This is done by organizing and managing the product backlog. The backlog is a prioritized list containing all the work items that need to be done to create a product.
- The team: With the guidance of the two roles mentioned above, the team is responsible for developing the product. A team member can be both a Scrum master and also a developer.

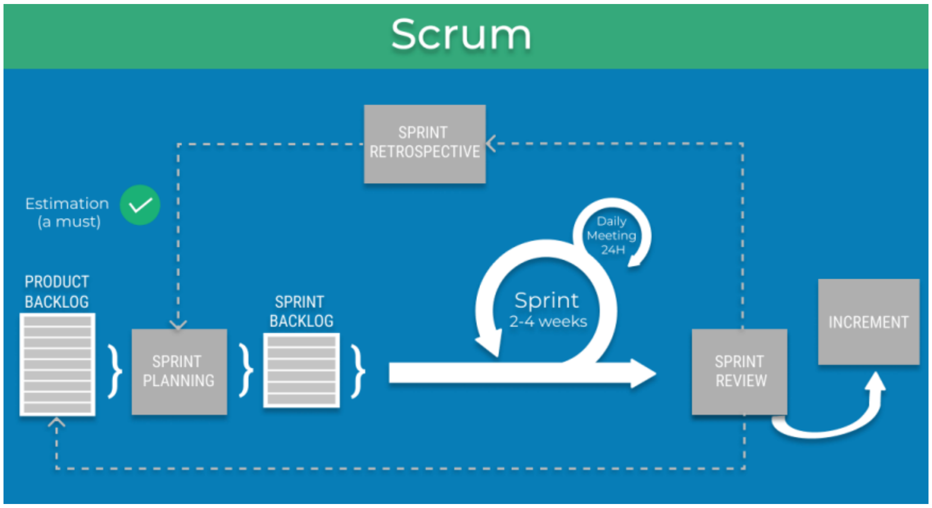


Figure 3.7: The Scrum process visualized(Schwaber & Sutherland, 2011, p.18).

In addition to these roles, several events need to be present in the development process. These events take place in and around a event that is referred to as a sprint. Sprints have a fixed length, most commonly under one month in length, that is repeated consequently. As mentioned there are several events surrounding the sprints, the relation of these events can be seen in Figure 3.7

Sprint planning

This event occurs at the beginning of each sprint and is performed by the whole Scrum team, it is also open to outside advisers. Sprint planning aims to answer three main questions:

- Why is this sprint valuable?
- What can be done this sprint?
- How will the chosen work get done?

To answer the first question, the Scrum team collaborates to define a sprint goal that communicates the sprint's value to the stakeholders. The second question refers to the amount of work included in the sprint. Typically the developers select items from a product backlog that they wish to include in a sprint. If any of the items need more refinement, this can happen now. Estimating how much work can be done and included in a sprint depends on the team's ability to predict and calculate how long a given task takes. Therefore, it is more common for experienced developers to give more precise estimates and more accurate sprint workloads. Finally, the last question requires the team to define when a backlog item is done. Typically this is done by splitting the item into several smaller tasks, as shown in Figure 3.8. The process results in a plan containing a list of tasks to be achieved in that sprint.

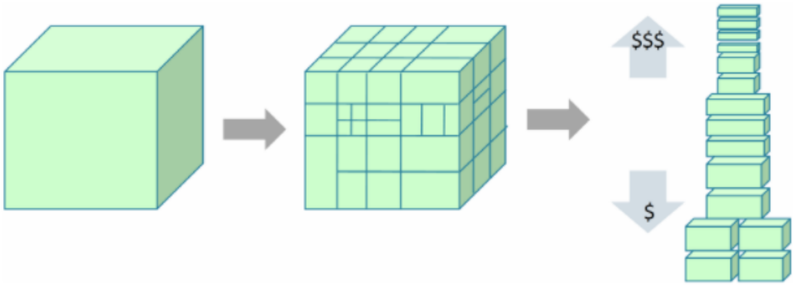


Figure 3.8: Scrum in a nutshell (Kniberg & Skarin, 2010, p.3).

Daily stand up

This is a much shorter and less complicated event than the sprint planning. Typically this event is held every day, takes 5 to 15 minutes, and is carried out by the developers. The goal is to look at the progress the team has made towards the sprint goal and keep the rest of the team up to speed on what each of the developers is working on. According to Schwaber and Sutherland (2011), daily stand ups improve communication, promote quick decision making, identify impediments, and because of this, they eliminate the need for other meetings.

Sprint review

As the name suggests, this event is for reviewing the current sprint and inspecting the outcome. This meeting is typically held on the last day of the sprint. The team members also present their work to key stakeholders as well as discuss the progress they have made toward the goal of the product.

Sprint retrospective

In this event, the team looks and plans for ways to improve the effectiveness and quality of the product and sprints. The typical procedure is for the team to collectively go through the processes that took place. It is also open for individuals to come with thoughts and suggestions for improvement. The most useful and impactful findings should be acted upon in the next sprint to improve effectiveness and quality.

3.4.2 Kanban

Kanban is an agile methodology that has been around for decades (Schwaber & Sutherland, 2011). The objective of Kanban is to visualize the workflow. This is achieved by splitting work into small pieces that can be written onto a card and added to a board. This board should be organized into columns with titles that explain the current state of that given task (Kniberg & Skarin, 2010). The purpose of these columns is to help teams and project managers to manage and organize their work, in addition to making it easier to maintain an overview of the project (Schwaber & Sutherland, 2011).

Kanban limits work in progress (WIP); this means that each column on a Kanban board has a maximum amount of items allowed. As shown in Figure 3.9, there is currently a limit on 2 WIP in the Ongoing column. This ensures that tasks are done and moved along the board before adding new ones. On the contrary, Scrum limits WIP per iteration, each sprint can only contain a certain amount of items, in Kanban, on the other hand, there are no sprints, but there can only be a finite amount of WIP items in each column. What this does, in theory, is make sure that the WIP in a column actually needs to be moved along the board before beginning with the next item, in Scrum, however, nothing is stopping the team from having all work items in a column. However, as mentioned before, Scrum is limited by the number of items in the current sprint, we can therefore conclude that while Kanban limits WIP directly, as can be seen in Figure 3.9 in the ongoing column, Scrum limits it indirectly (Kniberg & Skarin, 2010).

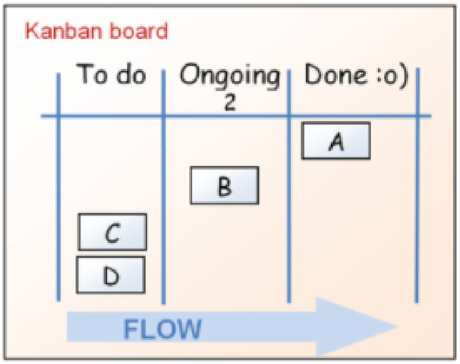


Figure 3.9: The Kanban board (Kniberg & Skarin, 2010, p.15).

Scrumban

Scrum and Kanban are both lightweight, agile methodologies. The many similarities allow for combinations of them, which is named Scrumban (Nikitina et al., 2012). Scrumban was created by Corey Ladas for the purpose of creating a way to transition from Scrum to Kanban. In his mind, Scrumban could serve as an intermediate stop before continuing to a full-on Kanban methodology. Scrumban takes the structured and predictable routines from Scrum and combines them with the flexibility that Kanban provides. A key element that Scrumban removes compared to Scrum is sprints, instead, it uses a Kanban board to manage work. This can possibly be time-saving as there no longer is a need to have sprint planning. As a result of removing sprint planning, the amount of estimation needed from the team is considerably reduced, as they no longer need to calculate how much they can do in each sprint. Instead, they are limited by the number of items they can have in each column or WIPs.

Chapter 4

Literature Review

This Chapter will review different literature relevant to our RQs. Our goal is to find issues in implementing Lean Startup, Design Thinking, and Scrum. Therefore we will review relevant research highlighting common issues in the aforementioned methods. In addition to this, we will also examine research about InnoDev and Joint Ventures, as these are relevant in this case study for understanding and later discussions on the implementation of the methods in ORG.

4.1 Critique of Lean Startup

Although Lean Startup has had rising popularity as an entrepreneurship strategy over the past decade, it has limitations and obstacles (York & York, 2019). Felin et al. (2019) criticized Lean Startup for pushing entrepreneurial activity to be based on a scientific approach, meaning that when it is correctly deployed, it might work; however, it leaves a larger room for mistakes.

One of the significant critiques Lean Startup has gotten is that it discourages radical innovation due to the high level of customer involvement. The build-measure-learn loop enhances an incremental innovation and could therefore misguide companies with more radical innovation and is not recommended in that scenario (Eisenmann et al., 2012; Felin et al., 2019). It is hard to argue that customer involvement is not important, but Lean Startup is criticized for encouraging it to start too soon. Involving customers early on assumes they know what they want and have a sense of viability for some future product. Felin et al. (2019) states learning is only as good as the startup's (or anyone's) ability to sense what they should look for in the first place. In some cases, this might be true for the customers, but you should not rely on it. The chase for validation from customers is not contributing to composing a novel hypothesis (Felin et al., 2019). This has been defended by Bocken and Snihur (2020) saying that Lean Startup promotes experimenting with reduced uncertainties at a relatively low cost and says it is the entrepreneur's hypothesis that decides the level of the radicalness of the new product or service.

Felin et al. (2019) does also criticize Lean Startup for lack of guidance regarding developing unique hypotheses so that entrepreneurs know what to look for in the first place. Creating a business model canvas is often associated with the envisioning step in Lean Startup (Bocken & Snihur, 2020). However, due to being an analytic and systematic tool, it reduces creativity at the initial phase (Snihur et al., 2021). Therefore, brainstorming, distant search, analogies, and other ideation activities are recommended. Arguably, Lean Startup is not a method for creating unique ideas but rather a method to ensure relatively low cost and increase the speed of the process (Bocken & Snihur, 2020).

4.2 Critique of Design Thinking

Design Thinking is a way of thinking and not a set of rules. Still, it is exposed to criticism, as there is room for interpretation. According to Iskander (2018), there are several faults with Design Thinking. Among several points, she claims that designers become a privileged entity in Design Thinking as they can act as gatekeepers of what ideas get included in the design process. The empowerment of the designer's role can lead to the narrowing of the potential for innovation as they can remove innovative ideas that they deem as irrelevant. In addition to this, the emphasizing part of the design process is also heavily affected by the designer's identity and experiences as they will relate more to issues that they themselves have experienced and can relate to (Iskander, 2018).

Design Thinking has also been criticized for not having a plan for the implementation and production of the generated ideas. In addition to this, it also does not provide any tools for further tracking and management of growth after a product has been launched. Because of these factors, one often needs a supplementary working methodology to accompany Design Thinking (Dobrigkeit et al., 2020).

As mentioned, Design Thinking is a way of thinking and not a recipe; the interpretation and understanding of how it should be executed can vary enormously. This can cause one person's criticism of Design Thinking to target a single interpretation of Design Thinking, but not all implementations of Design Thinking. Due to the lack of conspicuous criticism of Design Thinking, there is either a lack of research or a common belief that Design Thinking will always be beneficial if implemented with the correct mindset.

4.3 Critique and known challenges in Scrum

The Agile alliance bases itself on several assumptions and principles; by analyzing these assumptions, we can discover the limitations of Agile development. One of the assumptions is that customers are co-located with the developer team and are readily available when needed by the developers. There is also a reliance on face-to-face communication in Agile development (Turk et al., 2005). In this assumption, we can find a limitation regarding distributed development. If the development team is distributed over a wide or a global scale, it becomes difficult for the customer to be readily available. In addition to this, it can complicate the process of face-to-face communication. However, the last-mentioned factor has been reduced by adapting technological platforms that enable video communication (Turk et al., 2002).

A survey by Akif and Majeed (2012) has found and highlighted several limitations and problems with Scrum. The teams need to be highly agile since they often have short-term deadlines, this can cause developers to work extra hours or rush the development, leading to code quality deterioration. According to the survey results, 44% of the subjects stated that they have felt that managers have disrupted their teamwork. These disruptions include adding items and altering the current sprint, as well as asking about statuses such as one does in traditional software development life cycles (Akif & Majeed, 2012).

4.3.1 Backlog management

A central part of the Scrum methodology is the backlog and management thereof. Scrum has been criticized for lacking guidance concerning the backlog. This, combined with lacking Scrum training, can lead to struggles with the backlog.

50% of the team members surveyed lacked Scrum training. However, according to Akif and Majeed (2012), the main issue that can be found in Scrum is the constant need to be able to present something in a short matter of time. This leads developers to choose tasks in a current sprint that can be presented and often focus less on quality, leading to a build-up of quality-related items. In addition, performance-improving steps and tasks may also suffer from the same issue.

4.3.2 Estimation

Humans are known to be bad at estimating and often underestimate tasks. In 1977 Daniel Kahneman and Amos Tversky created a term for this, Planning Fallacy (Kahneman & Tversky, 1977). Their work claims that people often base their estimations on upcoming tasks and neglect historical data of similar tasks. Kahneman later expanded upon this term, stating that there are two key reasons for underestimating tasks (Kahneman, 2011):

- Failing to consider how long it's taken us to complete similar tasks in the past

- Assuming that we will not run into any complications that will cause delays

Since sprint planning relies heavily on estimation, we can likely find planning fallacy in Scrum. In addition to this, there is a lot of uncertainty in startups, which makes it hard to predict unforeseen obstacles and changes. According to Pulse of The Profession (2018), 25% of failed projects can be traced back to inaccurate time estimates. This demonstrates the importance of good estimates and the fatal consequences of getting them wrong.

4.3.3 Communication in the COVID-19 pandemic

Research by Brodnicki et al. (2021) delves deeper into the consequences of increased remote work under COVID-19. The research was conducted with 40 firms claiming to use Scrum. According to his research, remote communication has had a positive effect on the efficiency of the Scrum team, being more productive than co-located work.

Working remotely can however have a negative effect on the employee's self-esteem. This can be a result of isolation and lack of contact with the organization and external environments, as well as the loss of established bonds with colleagues (Brodnicki et al., 2021). A proposed solution by Brodnicki et al. (2021) is, therefore, to introduce practices that allow employees to do other tasks that are common to do in an office other than work, such as a coffee chat.

4.4 InnoDev

Several researchers attempt to create and test a hybrid model that combines Lean Startup, Design Thinking, and agile methodology. One of these models is InnoDev, which is an agile software development approach developed by Dobrigkeit, de Paula, et al. (2017) that combines Lean Startup, Design Thinking, and Scrum. The focus is on delivering innovative customer-oriented products and services in three defined development phases. All three approaches are relevant for software development but with different aspects of the process. Combining all three approaches have been argued to improve innovative software development, save costs, and increase the speed of development (Dobrigkeit et al., 2020). Individually, Lean Startup, agile, and Design Thinking have been criticized for lacking important aspects. For instance, Lean Startup does not cover how to derive the initial idea, Design Thinking lacks guidance on the implementation or production of the idea, and agile methodologies lack attention to the design aspect. In addition, neither Design Thinking nor agile methodologies offers guidance on tracking growth and how to scale after launch (Dobrigkeit et al., 2020).

InnoDev is a three-phase software development process as shown in Figure 4.1, the Design Thinking phase, the initial development phase, and the development phase. The Design Thinking phase accentuates activities mentioned in Section 3.3.1 to understand the customer needs. In this phase, Lean Startup activities support the validation of early ideas with testing, and Scrum activities support the project planning. As the understanding of the problem increases, the Design Thinking activities decrease, while development and business building increase (Dobrigkeit et al., 2019).

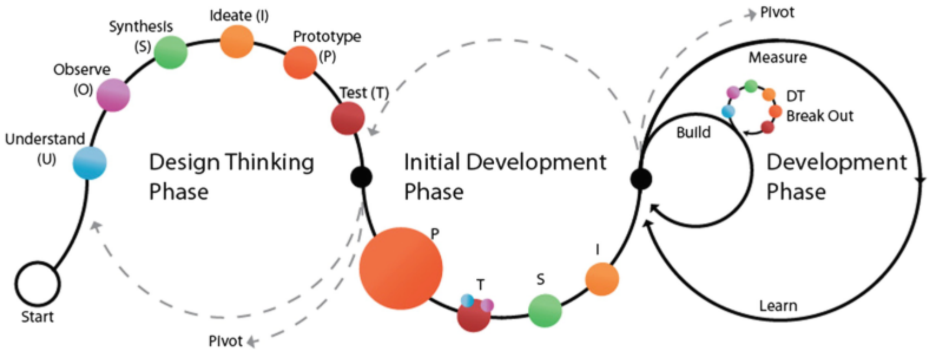


Figure 4.1: The three-phase InnoDev process (Dobrigkeit, de Paula, et al., 2017, p.326).

Combining Lean Startup, Design Thinking, and agile methods is nothing new and was recommended before InnoDev was assembled. However, InnoDev is a systematic and general model applicable for companies of different sizes. Due to being a relatively new method, more research is needed about the implementation to validate the benefits of the model (Dobrigkeit, de Paula, et al., 2017).

4.5 Joint Venture Startups

Many large companies are engaged in at least one joint venture. Some predominant reasons to create a new company as a joint venture instead of doing projects inside a well-established organization are that the project is; too large financially to handle the risks, it enables access to new markets and distribution networks, increases flexibility, and none of the companies possesses all the needed skills and resources to carry out the project alone (Killing, 2013). Joint ventures are reputedly well suited for learning and acquiring knowledge (Grant & Baden-Fuller, 2004) and provide a proper context for cooperative learning and the ability to create more value (Anand & Khanna, 2000). This increased opportunity to acquire knowledge increases the prerequisites to carry out innovation (Anderson et al., 2011).

Although there are multiple benefits to a joint venture, cooperating with another organization is not hassle-free. Communication, management, and expectations are some elements that need to be carefully monitored. Lack of trust between the owning partners, scarce information, cultural and managerial differences, and changes in the business or strategy are potential pitfalls a joint venture can encounter (Schillaci, 1987).

The first step to creating a joint venture is finding the correct partner to collaborate with where you are codependent and complementary to each other. Enlightening strengths, weaknesses, strategy, and expectations before deciding whether to collaborate or not will make it easier to choose the best partner. After deciding whom to partner up with, an agreement on a plan is crucial. Agreeing on the company's basic values, missions, and strategies will manage both partners' expectations and contribute to avoiding future conflicts. Still, the joint venture itself should be allowed to work independently without much interference from the partners, with someone all partners trust in charge. At the highest level, a board of directors should be established with representatives from both partners where decisions on strategy and open communication will happen to provide clarity and ongoing expectation management. Lastly, writing everything down and having a contract will decrease the chance of more significant conflicts and increases commitment (Gyenes, 1991).

Several non-monetary investments can be made when creating a joint venture. One of these is called Sweat Equity. Sweat equity refers to the action of owners or employees contributing to a company in other ways than cash. A typical example of this is to have a business rent out people at a discounted price to a company, and in return, they receive shares in the company. By doing this, they are increasing the value of the company. Additionally, they can make a profit if the company gets sold or becomes profitable (Kenton, 2020).

Chapter 5

Method

This chapter will describe the methods we have used to answer our research questions. Firstly, we will look at the methods used to collect the data and why these choices were made. After that, we will look at the data processing and the five-phased cycle. Then we will describe and reflect upon the reliability and quality of the data. Lastly, we will briefly explain how we cite the interviews in the next chapter and explain the ethical consideration surrounding data management.

5.1 Data collection

The data collected has primarily come from semi-structured qualitative interviews. This was our preferred way of collecting data to get a holistic view of the case from several perspectives. We have covered a longer period of time, so a quantitative study would not collect as accurate data as desired since the project has undergone many changes. Yin's recommendations from the book *Qualitative Research from Start to Finish* have had a great impact on the conduct of the study.

According to Yin (2011), there are four main methods of data collection activities in a qualitative study; Interviewing, observing, collecting and examining, and feeling. We have mainly focused on the first method mentioned above, interviewing. In addition, we have been present at the office for multiple weeks observing. We have interviewed eight people, all of whom have been involved in the project in the starting phase, either as board members, CEO, software developers, or designers.

The interview guide, added in English and Norwegian in Appendices A and B, was a guide but not a formal script as not all subjects mentioned were relevant to all interviewees. We aimed to avoid a tense interview setting but rather an open conversation where the main goal was to gather their story about ORG and their honest opinion about the choices that were made. We wished for the interviewees to lead the conversation, and we wanted to ask questions first and foremost to get them to elaborate. There were differences in how much the interviewees talked, and therefore, it varied how involved we were in leading the conversation further. In addition, there were some topics we were more interested in than others and would, first and foremost, ask follow-up questions relevant to them.

All interviews were held in Norwegian, except for one in English. Choice of language was based on the interviewee's preferences, and there was no language barrier.

The interviews were conducted in a private meeting room, where one of the students asked questions and had the conversation while the other timestamped and wrote notes. A qualitative interview follows a conversational mode, and as Yin (2011) states, it will lead to some sort of a social relationship, which happened in this case as well. We strongly believe that the interviewees have been open with us, given us detailed information, and stated their own opinions.

The interviews were recorded to make obtaining correct and detailed data for the analysis easier. The interviews lasted an average of about an hour each, which has given us significant amounts of data. In addition, we have been present at the office throughout the research period. Therefore, it has been easy to access and double-check information later on. This has been done through casual conversations in an open-plan office space at the office.

5.2 Data processing

A large amount of data has been collected, and it is essential to have a systematic way to efficiently sort and analyze the qualitative data. When doing qualitative research, we are not bound by any fixed methodology when analyzing the data. However, Yin (2011) expresses the importance of rigor and derives it into three precautions. (i) Checking and rechecking the accuracy of your data, (ii) Making your analysis as thorough and complete as possible rather than cutting corners;

and (iii) Continually acknowledging the unwanted biases imposed by your own values when you are analyzing your data (Yin, 2011, p.177).

There are numerous suggested ways to process the data and analyze it. We have chosen to follow the five-phased cycle from Yin (2011), which is illustrated in Figure 5.1. The first step of analyzing is compiling and sorting the data collected from interviews and documents provided by ORG and composing a database. Using this database, the next recommended step is to disassemble data by breaking it into smaller fragments. This procedure can be repeated many times as a trial-and-error process together with the third phase, which is reassembling the data. The goal is to reorganize the fragments into different groupings, analyze the data from different views, and look at relevant aspects. In the fourth phase, the reassembled material is used to create new narratives and interpret the data, and finally, the last step is to conclude and answer the research question. This is not a linear process, and how much time each phase needs differ considerably (Yin, 2011).

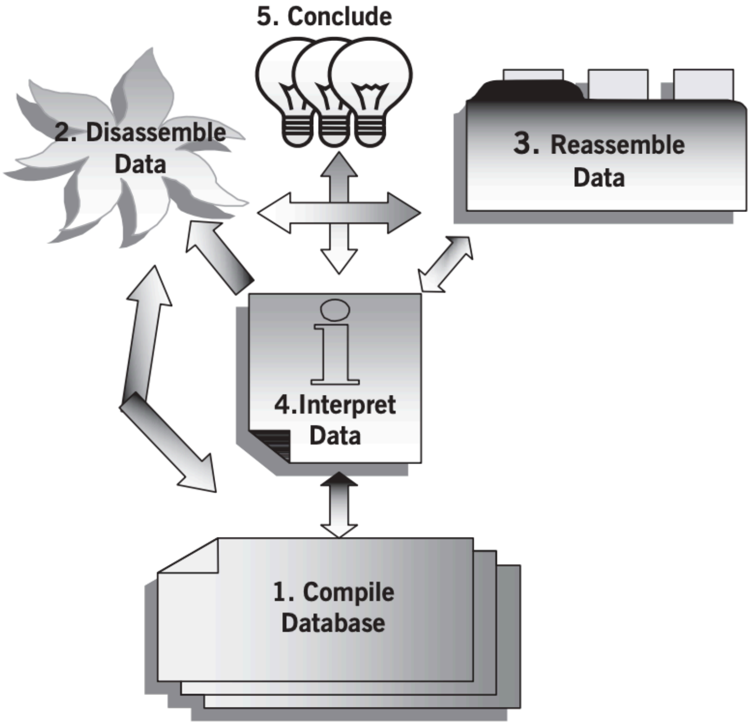


Figure 5.1: The five-phased cycle (Yin, 2011, p.177).

We did not use any software to analyze the data but created a database in Microsoft OneNote, which we carefully structured and processed by following the process described above.

5.3 Quality of the data

To assess the quality of the data, four criteria defined by Guba and Lincoln (1989) are examined to ensure the trustworthiness of this research.

- Credibility
- Transferability
- Dependability
- Confirmability

The results in this research are solely based on what the interviewees have told us, mainly in the interview, and some statements made in conversations later on. There is no discovered reason for anyone to lie or manipulate the data. While gathering data, we got to experience an openness from all participants and a wish to tell their stories. For that reason, we would think that if the interviews had been held again, the same results would have been obtained.

To ensure that the interviewees started the interviews on the same basis, we had a standard intro where we provided information and told them that our goal was to collect the history of the ORG with the main focus on Lean Startup, Design Thinking, and Scrum. In addition, we were clear that recordings were made and what they were to be used for.

The first things we established in every interview were their previous work experience and other relevant knowledge. There was an extensive range in how long and relevant experience they have. This is included in the assessment of statements that are not about the history of ORG, such as opinions about choices made.

When analyzing the data, we were careful not to manipulate statements but to stay close to what had been said. Most of the data has been translated from Norwegian to English, where we have, to the best of our ability, stayed close to the original statement.

5.3.1 Avoiding researcher bias

One of the students conducting this study is a part-time employee at *Partner A* and was hired as a software developer consultant to ORG for some time. Parts of the period overlap with the time period we have researched. Several measures have been taken to try to avoid research biases. This person has not been involved in the interviews but has been present to write notes along the way. The interviews were conducted by the other student who had no significant knowledge of the company or the people before this research period. In addition, we have been careful that the student with better knowledge of the company is only based on what has been said in interviews and other conversations with employees. This has been ensured by using references to statements and opinions throughout.

5.4 Ethical considerations and data management

This study was conducted after approval from the Norwegian Centre for Research Data (NSD). Before conducting the interviews, the participants received a formal letter and oral information about how the data would be used and stored. Both the interviewers and the participants signed a written consent. This study is anonymous, and therefore no personal data is exploited, but because of large amounts of data were collected, the interviews were recorded only for the purpose of researchers being able to transcript interviews later to increase the accuracy and be more present during the interviews. The participants were well aware and agreed to the interview being recorded, and the recordings were stored in the cloud with two-factor authentication. The personal data given in the interviews are not used. The participation was voluntary, and they could at any point leave and tell us to delete all collected data given from that person.

5.5 Citing the interviews

Further on in this thesis, statements made by the interviewees will be cited whenever necessary. They will be referred to as a Person in Department (PiD) to ensure anonymity while retaining relevant data about their field of knowledge and interests. For instance: *PiD A* said..., or if opinions are stated throughout the section, it will be cited at the end as (*PiD A*). Table 5.5 presents the departments in ORG.

Department	**Reference**
Board member	A
CEO	B
Software developer	C
Designer	D

Table 5.5: Participants profile

Chapter 6

Findings

In this chapter, we review and elaborate on the findings that have been made through interviews and conversations with employees and former employees of ORG. This chapter is structured into topics that deal with elements from Lean Startup, Design Thinking, and Scrum. Other notable findings that have influenced the development and implementation will also be addressed.

6.1 The structural attributes

When establishing ORG, *PiD A* said the structural attributes from Lean Startup, see Section 3.2.4, were highly influential. Startups are easier to manage and need less capital, which was the main reason for *Partner A* to not carry through their idea as an internal project but rather as a startup. Anyhow, they needed technical competence to implement the eSport platform, which *Partner B* possessed. *Partners A and B* agreed to finance the start phase of the startup with a total of 3 million NOK to finance the development of the MVP within a three-month time frame. The amount of money invested was a decision based on the concept *Scarce but secure resources* from Lean Startup. The agreement included *Partner B* being the long-term technical supplier providing both designers and software developers in the form of a sweat equity model. The CEO was intended to be neutral but was chosen by *Partner A*. The leader was chosen based on previous connections with *Partner A* and had earlier executed market analysis on eSport for *Partner A*.

An important aspect of Lean Startup is allowing the team in the startup to be autonomous, which the majority agrees has been the case. The only discovered exception within the first three months was when a board member had stronger opinions regarding development and handed over a project plan without involving the designers and developers in creating the plan. This plan included a time schedule for the development project before deciding exactly what the product was. Due to this plan being deficient, it was not used formally and had minimum impact on the project (*PiD D*). Other than that, the board had minimal guidance except for a Business to Business (B2B) solution requirement (*PiD A*).

There is a mismatch between the joint venture model where having consultants from *Partner B* and the structural attribute from Lean Startup regarding having a personal stake in the outcome. Only the CEO was a full-time employee in ORG, while both designers and software developers were consultants. Therefore financial reward systems were hard to implement. *Partner A* and the CEO did attempt to implement different financial rewards models, but there was never an agreement amongst everyone to implement them (*PiD A*). The personal stake does not necessarily need to be financial, as long as it increases motivation and makes a possible failure more painful. This can be done by creating a good company culture, but most people reported that the team spirit was poor due to a high level of conflicts and lack of social happenings for a longer period.

The composition of the team was decided by the budget (*PiD A*). Several designers mentioned they thought the team was too large early on. Lean Startup focuses on keeping the team to the minimum and recommends having a full-time representation in every functional department. *PiD C* states that it is a conflict of interest caused by the joint venture model where an owner also creates revenue by having consultants on the project. On the other side, a significant advantage *PiD A* stated is that ORG most likely would not exist if the situation were the same without the support from being a joint venture.

6.2 Management and crucial decisions

Multiple people, both from the design team and the software developer team, expressed concerns regarding the level of knowledge and experience about both eSport and managing a software development project. According to *PiD D*, it was done attempt to assist and increase the level of knowledge but indicated it was not received optimally.

The communication between the board and ORG went through the CEO. The CEO provided status updates and project estimates through formal and informal meetings. It was mentioned from *PiD B* and *PiD C* that the conversation with the board was affected by uncertainty if they were talking with an owner or a supplier as *Partner B* possessed both roles. *PiD A* said it was early on a lack of trust in the CEO due to underachieving within the first three months, misleading information provided during the meetings, and poorly performed estimation. As several mentioned, people, in general, are bad at estimation, and it gets even harder when the product is not defined well enough. It was also mentioned by several that they suspected the CEO promised more to *Partner C* than the rest of the board, which increased the expectations of *Partner C*. That led to tension and more difficult communication inside the board while adding pressure on the team. When the expectations were leveled out, communication within the board was reported to be good.

Inside the team, communication was perceived as varying. It is reported to be good communication between designers and software developers most of the time. Communication inside the software development team was influenced by some significant disagreements regarding the choice of agile development methods and technical decisions. While most discussions ended up with an agreement, it is clear that the software development team does not have the impression that it was open communication. Several meetings were held to improve communication, without any significant improvements reported. Changes in communication have only been reported after a change in the team's composition. But after the changes, everyone states communication has been open and good.

One of the major disagreements inside the software development team regarded the architecture and technical choices of the frontend. While the frontend developers wanted to implement micro-frontends and build a UI component-library from scratch, the backend developers disagreed as it would take more time and effort and was not suited for a startup with limited time. Due to this being a frontend decision, the frontend developers had a more significant voice regarding that choice. The choice of doing micro-frontends and UI component-library was on the basis of prioritizing high-quality over speed. While the backend team prioritized speed over high quality with the attitude of getting something done quickly to get it out in the market and tested and later go back and increase the quality. Both people from the software development team and the design team have pointed out that this has possibly been crucial for the project's development as the frontend development has been a bottleneck. Later on, the frontend was rewritten.

6.3 Development of the MVP

After the first three months, the product was nowhere near an MVP, and a new round of funding was necessary. A total of 10 million NOK was then invested, some from *Partner A* and *Partner B*, but the most significant amount came from a new investor, *Partner C*, which possessed a high level of domain knowledge. The money was estimated to be adequate to build the platform (*PiD A*). *Partner C* had stronger opinions regarding priorities and features in the platform, which was more centered toward professional gamers. The initial vision for the product was to create an eSport platform to make eSport more available for people with less eSport knowledge. Everyone agrees that the vision has always been the same, but interviewees mention that the vision was, to some extent, forgotten and unclear due to *Partner C*s involvement. There was no decision to pivot, but it gradually shifted into a professional gaming focus. Testing early on in the project revealed that the professional gaming market already offered good platforms, and the entry barrier was higher with more competition. In contrast, the non-professional eSport market had much more significant potential and less competition (*PiD D*). Despite that knowledge, the vision to some extent was forgotten, and the product got more similar to already existing platforms.

It is expressed by several interviewees that the decision of what to include in the MVP initially created disagreements in the team. The software development team said that instead of defining an MVP, they got a long list of features that corresponded to a full-fledged product and not just the minimum to be able to host eSport. Throughout the interviews, it was discovered that it is likely that the team does not share the exact definition of an MVP. When *PiD B* states that an MVP was defined, according to Ries (2011) definition, it was not the MVP but close to the finished product as it included more than the minimum to hit the market. The MVP followed when the vision changed, which evolved into an even longer list of features.

"It was a classic scope creep" -*PiD A*

According to *PiD D*, they designed the full-fledged product straight away and not an MVP. Due to the software developers getting the full-fledged product, the decision on what to include in the MVP was not clearly defined, but the design became the expectation. There was internal disagreement about whether the product should be made thoroughly with all minimum features at once or whether one should go for a more iterative process where one creates an MVP, according to Ries' definition, and later go back to improve and further develop functionality. Due to disagreements inside the software developer team, the CEO had a decisive say in the discussion, resulting in keeping the minimum features. The main reason for doing it that way was to ensure no shortcuts were taken during implementation. However, this is not conforming to the build-measure-learn feedback loop from Lean Startup.

6.4 Design Thinking implementation

The design process was started by two designers working together to create several user personas. To create these, they had several interviews and workshops with stakeholders and users. In addition to this, they had a gaming organization that contributed with domain knowledge. According to *PiD D*, they had a three-day Design Thinking workshop early on. Each day was dedicated to one of the themes; emphasizing, defining, and ideation. The participants consisted of all stakeholders and the gaming organization; there were 10-13 people present in total. The goal of this workshop was to find a vision and identity for ORG, with the main focus on a casual gamer (*PiD D*). The design team expressed that they thought this process worked well and were happy with the outcome.

After the workshop, they started creating a prototype based on the collected knowledge from the workshop, validated wireframes, and interviews. According to *PiD D*, it took approximately four months from the idea to a prototype, but the initial work on the idea started before the company was founded. As mentioned, the prototype included all features and was a full-fledged prototype.

6.5 Testing and feedback

An important aspect of Lean Startup and Design Thinking is testing the product and getting feedback from potential customer groups. At the beginning of the project, the concept mapping was done. According to *PiD D*, a systematic way of handling feedback was implemented through two different types of interviews, exploratory and validating. The exploratory interviews have aimed to get an unbiased approach and explore the problem. In contrast, the validating interviews aimed to validate the concepts made to solve the problem. The interviewees were mostly friends and acquaintances, family members, and professional gamers. Maze testing was also performed on the prototypes. *PiD D* states that they should have had more focus on exploratory interviews but says that it was more accepted in the project to work solution-oriented instead of problem-oriented, while *PiD D* believes both must be present.

The first prototype was, according to the designers, well-validated and targeted toward lower-level eSport. As deadlines started approaching and the technical product was far away from the prototype, changes to the prototype were made to make it more achievable for the software developers *PiD D*. Those changes seemed to be rushed, and on some occasions, time and effort were put into developing features that later were scrapped because it was not tested well enough or the design was changed later on *PiD A*. A problem several designers pointed out is the fact that the software development started two to three weeks after the workshop where they found their vision. According to the designers, this gap was too small and resulted in stress and not enough time to validate and iterate on designs.

Eventually, it was decided to scale down the prototype to a version they called 0.5. This was the minimum version possible to deliver by the deadline. As this was a decision made in order to reach a deadline, there was minimal time for testing, and 0.5 ended up never getting tested. *PiD D* states that it seemed like it was a fear to test the 0.5 version as they did not believe it would be validated.

> “The validated prototype was thrown out in a panic choice” -*PiD D*

When it comes to the technical product, *PiD C* says that because the MVP was not clearly defined, they did not know when to test it and therefore ended up waiting too long. Overall, most people say that testing was initially sufficient, but as the pressure increased due to deadlines, changes were made without validating them. A couple of the team members say they advised the team that testing should be a greater focus later on, but it did not significantly impact future decisions and priorities.

6.6 Implementing Scrum

Through interviews with the software developers, it became clear that Scrum was implemented when software development began. Even though Scrum was implemented, several interviewees claimed that the development process ended up looking more like a waterfall model. *PiD D* and *PiD A* both claim that one of the causes was the fact that the software development began too early. Due to Scrum turning into waterfall, the team felt the need to hold a workshop to determine the future course and choice of work methodology. The workshop was held a couple of months after the development of the product had begun. After a thorough discussion, it was concluded that they were going to continue using Scrum, even though members of departments A and B protested that it has a lot of overhead in the form of meetings and ceremonies as well as demanding a lot of knowledge and skills from a product owner. *PiD A* claiming that they would most likely not be able to stick to Scrum for long and would end up with Scrumban or Kanban instead.

Scrum has many ceremonies and a consistent structure. Everyone mentioned that Scrum contributed to organizing the communication in the project because of the regular ceremonies and meetings that are a part of the Scrum methodology. It also helped maintain regular communication when COVID-19 restrictions occurred and, in general, when people had home offices, which was frequently used by some of the team members.

However, a drawback of the rigid Scrum structure is that it is not allowed to add new tasks once a sprint has begun. As the team faced changing deadlines and regular changes in specifications, which we will elaborate in 6.8, the team diverged from Scrum towards a Scrumban methodology.

6.7 Backlog management

As mentioned in Section 4.3, there is little guidance regarding the management of the backlog. This became evident when 5 out of 8 felt that the backlog and management were lacking. The product owner at the time also agrees that the backlog was an issue, as the content was not up to the expected standards of the development team. *PiD C* says that the backlog contained not just what was necessary for the MVP, which was poorly defined, but also everything wanted for a final product. Several team members also claimed that the list of features in the backlog was not prioritized sufficiently.

After recognizing the backlog issue, the team chose to rotate the roles such that one of the software developers became the product owner. The reasoning behind this was to achieve higher quality items in the backlog as well as a more organized structure as the developer had a better understanding of how it should be done. Even though the right to add new items to the backlog was with the new product owner, members of the development team reported that items were still being added directly to the backlog without the product owner's knowledge. This conflicts with the roles and responsibilities that come with the roles as defined in Section 3.4.1. According to multiple team members, the change of product owner did improve the quality of the items in the backlog.

According to *PiD C*, the change of the product owner role was not without problems. The person who obtained this role already had several other roles such as CTO, developer, and now also product owner. Therefore, the new product owner had limited time for each of the tasks, and the new responsibility of organizing and increasing the quality of the backend was a time-consuming task. *PiD C* state that this led to increased stress and pressure that could have been avoided if the CEO had been able to perform this role as intended instead.

6.8 Deadlines and estimation

Some of the factors that pushed the team towards Scrumban were deadline changes as well as changes in specifications. 5 out of 8 interviewees answered that the deadlines had negative effects. Some of the negative effects mentioned were; their ability to deliver, team morale, and caused changes in the current specifications and the MVP. As mentioned in Section 6.5, because of sudden deadline changes, design aspects had to be cut down. This resulted in the need for work done by developers to be thrown out and start again. When these deadline changes appeared in the middle of sprints, it resulted in new tasks being added to a sprint. This caused the sprint to suddenly contain more work than the team had estimated and agreed upon in the sprint planning ceremony. The result of this was a sprint board that was too large, and work items kept being transferred to the next sprint. As this process continued, the Scrum board looked increasingly like a Kanban board.

> "New things added during sprints or deadlines that were changed caused sprints to fail " -*PiD C*

A factor that amplified the situation surrounding the deadlines is the fact that several team members were unaware of why and where the deadlines came from. *PiD D* states that deadlines seemingly were drawn from loose air and that the communication surrounding them was not good enough. In addition, several states that these deadlines were created based on rough estimates made by the software development team, which were presented, without further rectifying, as strict deadlines to the board by the daily leader. *PiD C* states that these estimates were not processed enough to use these as deadlines. However, after the team failed to reach the deadlines multiple times, management realized that something needed to change, this caused the team to reflect as well as adopt a new strategy. This displays the team's ability to adapt and change based on experiences and situations (*PiD B*).

Chapter 7

Discussion

In this chapter, we consider the key findings from our research and discuss them to be able to answer the research questions. The discussion will mainly consider how the implementation of Lean Startup, Design Thinking, and Scrum went in this case study, and further on look into previous related research and compare issues and situations.

7.1 Key Findings

7.1.1 Lean Startup

The key findings regarding the usage and implementation of Lean Startup are that attempts have been made to implement it. Still, only individual elements from the method have been used for several reasons. Therefore, it can not be claimed that Lean Startup has been used to a great extent in this case.

Some structural success factors, mentioned in Section 3.2.4, as having independent development authority and scarce but secure resources have been utilized. But having a personal stake in the outcome was not implemented in any financial manner due to ORG having consultants and not full-time employees. Critical success factors, as mentioned in Section 3.2.3, includes the development of the MVP, using customer feedback, and pivoting if needed. Defining the MVP has been an important topic in this case study. There are many indications that the first version of the MVP was too large and poorly defined for the software developers. This version of the design prototype, however, was well tested and verified by the customer group. Later it was reduced to meet deadlines, and the new design prototype, named version 0.5, was not adequately tested. Therefore, this change was at the expense of the critical concept in Lean Startup, customer feedback. Regarding the testing and feedback of the developed MVP, it was tested very late due to poor MVP definition. The vision for the product has remained, although it has not always been apparent and in focus along the way due to influence and promises to *Partner C*, meaning no pivoting has been necessary.

7.1.2 Design Thinking

The key findings regarding the usage and implementation of Design Thinking are that the Double Diamond Design Thinking model was carefully used at the beginning of the project. The process of creating the design prototype started two to three weeks before the software development. It could have been more exploratory, but user personas were made, and the user testing and verification seem to be sufficient.

A full-fledged product was delivered to the software developers, that later on had to be reduced due to deadlines. When scoping down the entire product to the 0.5 version, there are many indications that Design Thinking was neglected. In the process of scoping down, designs were delivered to the software developers that had to be scrapped due to not being user-friendly enough. The design was also affected by *Partner C's* involvement, which led to a minor derailment focusing on professional gaming, conflicting with the initial vision.

7.1.3 Scrum

At the beginning of the development process, ORG implemented Scrum, however it was reported that this implementation resembled a waterfall process. Therefore, the team decided to have a workshop to address the issues they were facing. They decided to refocus and give Scrum another go, even though they faced opposition from the management team. The overall implementation of Scrum was reported to have led to an increase in communication and structure in the team.

The team struggled on several points with the Scrum implementation and guidelines. An especially crucial point of contention was the management of the backlog, as the current product owner did not possess the required knowledge to create and manage the product in a manner that the team expected and required.

This became clear as the backlog was not prioritized correctly. To address this issue, they appointed a new individual as the product owner. However, items were being added to the backlog by others than the new product owner, which is also in conflict with the Scrum guidelines.

The team also faced issues due to deadlines and estimations. The estimations of the tasks in the sprint were poorly done but used as set deadlines. Changing deadlines led to a reprioritization in the middle of the sprint, which resulted in items being added to ongoing sprints. This resulted in items being transferred from one sprint to another as there were more pressing items to work on. This turned the Scrum implementation into an implementation that resembled Scrumban.

7.2 Partly implementation of Lean Startup

In this case, some elements of Lean Startup have been implemented, while others were harder to achieve for various reasons. As mentioned in Section 3.2.2, Lean Startup is a method and not a process, meaning there are no direct rules on how to use the principles. Still, guidance is easily accessible, and modifications can be made to adapt to one own needs and situations.

First and foremost, having sufficient knowledge about Lean Startup is essential for the company to be able to implement it and get its benefits. As Ries (2011) states, entrepreneurship is management, and being flexible and learning-oriented is essential. Both statements and actions in ORG indicate a lack of knowledge when decisions were made to be compliant with Lean Startup. Defining the MVP is an important aspect that has not been optimal in this case. This is a well-known challenge in Lean Startup. While a review on Lean Startup Lizarelli et al. (2021) focuses on the difficulties of creating an original and sellable MVP, this case enlightens the difficulties of defining an MVP within the frame of time and money available.

Even though MVP is an abbreviation of “Minimal Viable Product”, the focus could easily shift into seeing how much we can add rather than what is the minimum to create value within the given time frame. That seems to have been the case here as well. There were disagreements regarding what to include in the MVP, and the decision was to add as much as possible. This requires some sort of estimation, which is shown to be hard, especially in software development (Molokken & Jørgensen, 2003). In this case, some rough estimations were done, but they were far from accurate. High expectations increase the chance of underestimating the effort required (Jørgensen & Sjøberg, 2004). Added pressure from deadlines, in this case, is likely to have impacted the estimation.

Looking at the suggested Lean Startup process in Section 3.2.2, the build-measure-learn feedback loop was never initiated with a working product within the time frame this case study undertakes. One of the core principles of Lean Startup, innovation accounting, focuses on the importance of measuring, prioritizing, and setting milestones. This was not clearly present in this case, which may increase the difficulty of defining the MVP and starting the build-measure-learn feedback loop.

The initial prototype made by the designers does, to some extent, have a build-measure-learn feedback loop but specifically follows the Double Diamond approach. It is a similar approach as it is an iterative process focusing on the user's needs and verifying hypotheses on how to achieve market fit. The build-measure-learn feedback loop may bring some unclear and contradicting advice, as Lean Startup's primary goal is to increase speed and decrease cost. Still, it encourages redesign and then validation. This may lead to the costly process of redesigning more than necessary. Incorporating the build-measure-learn feedback loop into Design Thinking and not the software development process may be a cheaper option. Testing and verifying the developed product should still be encouraged, but with minimal risk of pivoting due to a good Design Thinking process. Methodologies combining Lean Startup, Design Thinking, and Scrum into a hybrid model as InnoDev, have a similar approach doing the Design Thinking phase first and then implementing the build-measure-learn loop in the development phase.

Since Ries (2011) introduced Lean Startup, it has developed into a buzzword used by both startups and larger organizations. The increased number of organizations stating it is "Lean" or following Lean Startup indicates it is a beneficial and effective methodology (Gray, 2021; Yordanova, 2020). However, it may happen that one states they are Lean or a Lean Startup without being it, to be trendy. Looking into all the elements of Lean Startup, it demands knowledge and dedication to implement successfully. Whether most elements need to be present to benefit from the Lean Startup methodology is a subject of interest, as this case cannot conclude with the results of implementing the methodology.

7.3 Deadlines and collaboration affecting the Double Diamond model

The implementation of the Double Diamond model based on Design Thinking, compared to Lean Startup and Scrum, was talked about more positively. The design team showed knowledge and interest in Design Thinking and the Double Diamond model. In the beginning, every Double Diamond model element was used; discover, define, develop, and deliver. Later on, the discovery part became less of a priority. This part is important as it discovers the issue the product will solve. As mentioned in Section 6.3, the vision was, to some extent, forgotten due to *Partner C*'s professional gaming involvement. The continuous discovery could be a beneficial action helping to maintain focus on the user's needs. In addition, Jolak et al. (2021) suggests that co-located teams spend more time exploring the problem space than distributed teams, meaning that the COVID-19 pandemic may have had an impact.

The most visible issue with the usage of the Double Diamond model, in this case, was the lack of time to carry it out. The design team had a three-week lead on the software developer team, adding pressure to deliver the prototype quickly. When the prototype eventually was scaled down to the 0.5 version, the deadline forced a fast process, not giving the designers time to discover and validate. Literature on time management with Design Thinking teams is still scarce (Häger, Uflacker, et al., 2016) but is an important factor in this case. Data clearly shows an increased return on interest when making human-centered methods a priority (Rae, 2014). Still, there is no clear literature on why testing the design may be neglected under time pressure.

The software developers were minimally included in the design process throughout the development. Some communication and collaboration are visible but may not be optimal. This could increase the time from the design being developed until the software developers are aware of the effort expected by the design. The feedback time from the software developers on the achievability of the design increases, not encouraging a fast feedback loop.

7.4 Reviewing Scrum implementation issues

As mentioned in Section 6.6, the team early on adopted Scrum. However, it was stated that the work methodology felt more like a waterfall model than an iterative Scrum process. A possible cause for this presented by the team is that the developers started too close to the designers. This lasted almost three months before they decided to look closer at the issue and address it with a workshop. Three months can be viewed as a long time span for a startup, where time to market is a crucial success factor. The Scrum methodology is agile and iterative; therefore, it should be able to react to change effectively. It is possible that the accidental shift towards a waterfall model delayed the team's response time.

Another factor that could have caused Scrum to not work as intended could be that the first designed prototypes the software team received were almost a complete product. This could lead to a waterfall approach instead of an iterative approach as one wants in Scrum and other agile methodologies. In this case, it appears to be an unfortunate situation with the design team not having enough time to fully finish the prototype but still deciding to deliver a finished prototype and not follow an iterative process with the software development team. It is highly believable that the finished prototype has influenced the choice of what to include in the MVP. Using Scrum with Lean Startup and Design Thinking may have been optimized if hybrid models that combine this workflow, such as InnoDev, influenced the process.

According to Akif and Majeed (2012), it is common for developers under pressure to typically sacrifice quality, performance-improving steps, and tasks in sprints. We argue that time pressure is one of the main causes for the conflicts regarding the technical choices, as the chosen technologies such as micro frontend and a custom UI library are more time-consuming than other solutions. The team was split on these choices as some regarded them as subpar choices because they needed speed over future scale-ability, while others preferred quality and future scalability now over speed. This is possibly also an influencing factor for some of the internal conflicts in the team.

7.5 The importance of Scrum knowledge

The lack of training and knowledge about Scrum has most likely affected the Scrum implementation. Throughout the interviews, it became clear that some of the most crucial members of the team lacked training or knowledge of Scrum. It was also reported that some team members felt that they lacked the mandates to perform the roles they had been appointed to. This could be caused by the insufficient amount of knowledge regarding the different roles in Scrum.

In addition to this, it was also unclear who in the team had to bear the responsibilities of the product owner role, as the intended product owner did not have the capacity to create work items that meet the demands of the development team. It is reasonable to believe some issues could have been avoided with a more thorough training and competency development in Scrum. The fact that the intended product owner could not take this role led to cascading effects, as one of the backend developers had to take the role in addition to his other roles as technical lead and developer. It was reported that this increased stress and pressure on the developer in question. An additional cause for this was a constant need to provide and prepare tasks for the development team. This was a time-consuming task as the items in the backlog were not up to par and had to be worked on before passing them on to the developers.

7.6 Backlog management

A direct consequence of the lack of Scrum knowledge in the product owner role was the management of the backlog. Akif and Majeed (2012) claims that there is a lack of guidance regarding managing the backlog in Scrum. This appeared inherent in this case. As mentioned, the product owner role was moved from the current CEO to a software developer. However, the CEO continued to add items to the backlog. As we mentioned in Section 4.3, this can be a common issue.

The Scrum backlog should be a prioritized list containing all the work items to be completed to create an MVP or product. As we have mentioned, there have been issues with the management of the product backlog. In addition to new items being added at random without a mandate to do so, the prioritization was wrong, as it was not present, according to our findings. One would expect to find features and tasks in a final product that is not typically part of an MVP. Once again, we state that this is not a rare mistake to make as it is known that Scrum lacks guidance on backlog management.

7.7 Estimation and agility of Scrum

A central part of sprint planning is doing estimation. This is crucial for the sprint's success, as underestimation of task complexity results in not being able to complete the tasks in the current sprint. However, underestimation of tasks is a common issue, as mentioned in Section 4.3.2, planning fallacy is a term used to describe this. The team suffered from underestimating in several parts of the project, as well as in the sprint planning. The planning fallacy could have played a central role in the early phases, where large portions of the code were thrown away due to new deadlines. This appears to be a common phenomenon; as we mentioned earlier in Section 4.3.2, 25% of projects fail because of failure to estimate tasks correctly.

As mentioned in Section 6.6, the team eventually transitioned from Scrum to Scrumban/Kanban, as shown in Figure 7.1. Work items were being moved from one sprint to another, and items were being added mid-sprint; these actions contradict the rules of Scrum. Therefore, we see this transition to Scrumban as a natural choice. In addition to the aforementioned issues, a core difference between Scrum and Kanban is continuous delivery versus time-boxed sprints. Since ORG is a startup, deliveries and deadlines are created on short notice, and a Scrumban approach would yield a more agile and flexible development cycle. According to Ahmad et al. (2016), software companies are increasingly moving from Scrum to Kanban. When the team decided to shift away from a full-on-scrum to another agile methodology, they had received some benefits from the rigid ceremony structure from Scrum. One of these benefits is increased communication which many indicated would not have happened if they had not used Scrum first. Therefore the move to Scrumban seems natural.

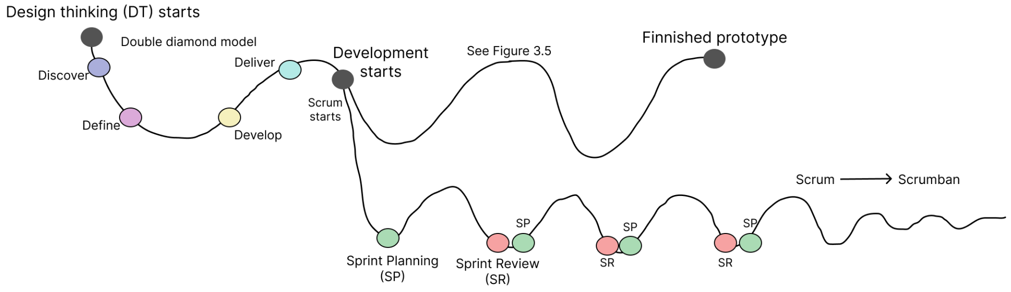


Figure 7.1: Illustration of what the process in the start-up phase may have looked like. This is not an exact representation.

7.8 COVID-19's affect on Scrum

Several of the interviewees mentioned that the implementation of Scrum benefited the startup in creating structured meetings and communication as it has many ceremonies requiring team members to attend. We believe that the timing of these routines and ceremonies benefited the ORG in challenging times with COVID-19. The team had routines and therefore had regular meetings even though members of the team were working from home. This theory can be supported by the fact that when asked what effect the team felt from COVID-19, many members claimed that it had little effect on the overall development.

A challenge that arises in Scrum is the need to be co-located (Turk et al., 2005). ORG has proved to overcome the need to be co-located, as several developers were working abroad. In addition to this, there was a lengthy period of time where developers and management were forced to work from home offices due to COVID. As mentioned in the previous section, the team reported that they did not feel that COVID affected the development to a large extent. We argue that one of the reasons they were able to overcome this challenge is due to the progress and innovations in terms of video communication software such as Microsoft teams and zoom and the use of Scrum contributing to regular ceremonies and meetings. This theory is supported by the works of Andres (2002) that claim that video communication can lead to superior decision quality compared to face-to-face meetings. However, it is also mentioned that group support and group well-being functions are more present in face-to-face meetings. It is important to note that the research was conducted in 2002, and video communication tools have changed enormously.

As mentioned, the team did not feel that COVID-19 affected the development. A research paper by Brodnicki et al. (2021), mentioned in Section 4.3.3, supports their opinions and claims that the overall efficiency in Scrum projects was improved under the pandemic compared to normal in-office work. In the same paper, it was stated that the lack of normal office activities other than working had led to lower self-esteem as well as an increased feeling of loneliness and isolation. The product owner at the time stated that they felt alone in the decision-making processes. These feelings could be affected by COVID-19 and could have been improved upon by the proposed solution in Section 4.3.3.

7.9 Joint venture affecting the implementations

The structure of ORG, being a joint venture and using consultants from *Partner B* have in some essential ways affected the project. Being both an owner and a supplier, as *Partner B*, may be an economically beneficial model. But it entails new situations and increases complexity when implementing work methodologies in the joint venture. Having consultants rather than full-time employees allows the company to have a more flexible workload. At the same time, the consultants do not have any personal stake in the outcome, which was stated to be an issue throughout the project because their salary is independent of the startup's success. Intrinsic motivation may make up for some of it, but it is highly variable. The startup mentality and the fast and cheap development are harder to achieve when there is no personal stake in the outcome. There was also an unclear expectation from the stakeholders, which can be both positive and negative. It allows agility and creativity inside the startup, which is important in the initial phase. At the same time, this freedom depends on high-quality leadership inside the company. The CEO being the only full-time employee at this time increases their responsibility and adds pressure. If more people were full-time employees, multiple people would have the mandate to impact decisions. Having a group rather than individuals reduces the chance of error in judgment and reduces various biases (Larrick, 2004).

Some of the technological choices for this project decreased the development speed but were chosen based on increased scalability and quality. At the early stage of the product development, ease of development and speed is important prior to finding product-market-fit (Hultberg, 2021). Choosing technology known to demand more time and skills, which to some degree happened in this case, contradicts what a startup generally wants. The choice made was characterized by individual preferences and a consultant focusing on quality. This might be a result of the startup mentality not being fully present.

A significant advantage of ORG being a joint venture is a higher threshold for terminating the project as they have a lot of support. It is impossible to tell if the same situations would occur in an independent company, but most likely, they would have had more full-time employees from an early stage. As a startup not owned by larger companies, it could also be able to apply for funding through governmental systems. Still, it would not be any guarantee the project would survive as long as it has, as most startups are likely to fail.

7.10 Iterating methods

Overall, the methods implemented are based on iterative processes. The iterative approach should be able to help the team react quickly if changes should be made. For various reasons, the iterative part of the methods has disappeared gradually or never started. The problems in Scrum with the backlog management meant that the sprints were not finished, eventually looking like the waterfall method. Lean Startup's iterative process never started because the MVP was large and was not completed within the period we are looking into. And Design Thinking, which could have had even more focus on continuous exploring and did not have time to validate the changes in the event of significant design changes. A similarity with the implementation of the methods leading to iterative processes stopping was the general focus on creating a full-worthy product immediately and not on dividing it into smaller pieces. In summary, the iterating events became more one-time events.

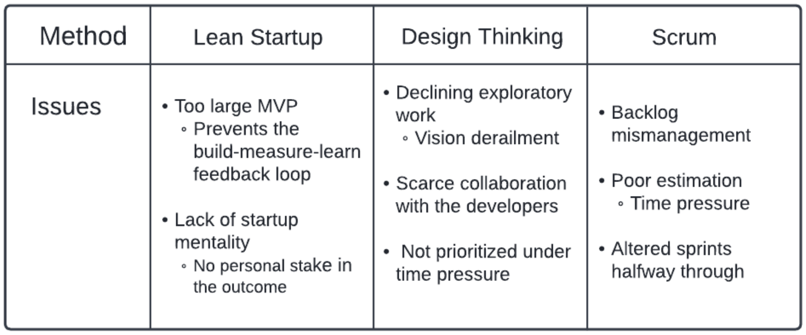
Chapter 8

Conclusions

This chapter will conclude regarding the three research questions, answering what issues ORG encountered when implementing Lean Startup, Design Thinking, and Scrum.

- RQ1: What were the issues for ORG when implementing Lean Startup?
- RQ2: What were the issues for ORG when implementing Design Thinking?
- RQ3: What were the issues for ORG when implementing Scrum?

Eight semi-structured qualitative interviews have been conducted, in addition to observations and conversations along the way, in order to answer the research questions. Our conclusion of the RQs is summarized in Figure 8.1 and is elaborated in the following sections. Finally, we will recommend further relevant research.



Method	Lean Startup	Design Thinking	Scrum
Issues	• Too large MVP ◦ Prevents the build-measure-learn feedback loop • Lack of startup mentality ◦ No personal stake in the outcome	• Declining exploratory work ◦ Vision derailment • Scarce collaboration with the developers • Not prioritized under time pressure	• Backlog mismanagement • Poor estimation ◦ Time pressure • Altered sprints halfway through

Figure 8.1: A summary of the conclusion.

8.1 RQ1: What were the issues for ORG when implementing Lean Startup?

The most apparent issue ORG has had with implementing Lean Startup is defining and developing the MVP. The MVP was not defined as an MVP but as a complete product which meant that the Lean Startup process with the build-measure-learn feedback loop did not start within the time period, we have examined. This indicates a lack of knowledge regarding Lean Startup. Because the process got interrupted, several essential elements of Lean Startup are not included. Another issue was that some decisions were made without considering fast development, which is the goal of Lean Startup. By having hired consultants from one of the investors, the company's structure may have contributed to limiting the startup mentality in the company and enhanced a conflict of interest since the supplier financially benefits from delivering consultants for a more extended period. The consultants did not get any major personal stake in the outcome. Hiring consultants is more expensive than hiring full-time employees, and in a startup with limited funds, it will shorten the time frame they have available. Scarce but secure resources may have been a focus initially, but having too many consultants early on has been an issue.

8.2 RQ2: What were the issues for ORG when implementing Design Thinking?

Design Thinking is a way of thinking and not a method, so this conclusion is based on the implementation of the Double Diamond model. Higher knowledge of the model leads to better implementation, which was the case compared to Lean Startup and Scrum implementation. But higher knowledge was not able to avoid Design Thinking being less prioritized under higher time pressure. There is reason to believe that it is common for testing and validation to be among the first to fail under high time pressure. In this case, developing poorly tested prototypes, as the 0.5 version was, has led to more work later. It might pay off to keep testing and validation even under high time pressure, as it costs less to make changes to the prototype than to the product. In addition, it has been difficult to maintain a focus on continuous exploration, which leads to the implementation partly be-coming a one-time event and not iterative. A possible consequence of this may have been that the focus on the main problem they wanted to solve derailed along the way with the involvement of one of the partners.

8.3 RQ3: What were the issues for ORG when implementing Scrum?

The main issues we have discovered in concerns to Scrum are backlog mismanagement and poor estimations leading to time pressure. The mismanagement of the backlog may have been caused by the lacking knowledge about Scrum in the product owner role. A large backlog created more pressure on the team. The total pressure may have intensified some conflicts and decreased the team spirit. Sudden deadline changes led to sprints being altered halfway through, which resulted in looking like a waterfall method and later in a slow, maybe unintended, transition toward Scrumban as work items were transferred from one sprint to another. In addition to this, it seemed the team often suffered from underestimation. The estimates that were made were not processed enough to be used as reasonable estimates but were presented as such. Finally, we also saw that COVID-19 may have played a role in the execution of Scrum as it possibly led to more isolation of the different members of the team. Even though Scrum contributed to regular communication, it is time-consuming, meaning that it may not be the best long-term work methodology under high time pressure and in a startup trying to develop fast.

8.4 Further recommendations

As this is a case of using multiple methods separately, a suggestion is to compare this to a set hybrid model such as InnoDev, to see what advantages and disadvantages each approach may have. As this is a single case study, another suggestion is to conduct multiple similar case studies to see which issues are most common, to eliminate the risk of one-off cases. Many of these methodologies overlap and have similar basic principles. It would be interesting to examine how one methodology can benefit another.

As concluded in this case, Design Thinking processes and general testing of the product have been neglected under high time pressure. It would have been an interesting research topic as relevant literature is scarce to know whether it is a common occurrence and why it happens.

A lot has happened with ORG since the time period we have looked at; issues have been addressed, the company has changed its structure, and new working methods have been adopted. Therefore, it would have been of interest to carry out a new case study of ORG that addresses the changes that have taken place and look into the transition and use of the new methodologies.

References

Ahmad, M. O., Kuvaja, P., Oivo, M., & Markkula, J. (2016). Transition of software maintenance teams from scrum to kanban. *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 5427–5436.

Akif, R., & Majeed, H. (2012). Issues and challenges in scrum implementation. *International Journal of Scientific & Engineering Research*, *3*(8), 1–4.

Anand, B. N., & Khanna, T. (2000). Do firms learn to create value? the case of alliances. *Strategic management journal*, *21*(3), 295–315.

Anderson, A. R., Benavides-Espinosa, M. d. M., & Mohedano-Suanes, A. (2011). Innovation in services through learning in a joint venture. *The Service Industries Journal*, *31*(12), 2019–2032.

Andres, H. P. (2002). A comparison of face-to-face and virtual software development teams. *Team Performance Management: An International Journal*.

Balaji, S., & Murugaiyan, M. S. (2012). Waterfall vs. v-model vs. agile: A comparative study on sdlc. *International Journal of Information Technology and Business Management*, *2*(1), 26–30.

Bocken, N., & Snihur, Y. (2020). Lean startup and the business model: Experimenting for novelty and impact. *Long Range Planning*, *53*(4), 101953.

Boynton, A. C., & Zmud, R. W. (1984). An assessment of critical success factors. *Sloan management review*, *25*(4), 17–27.

Brodnicki, K. et al. (2021). Remote communication in scrum teams-a covid-19 preventive measure or work time optimisation? *e-mentor*, *92*(5), 81–90.

Canedo, E. D., Pergentino, A. C. D. S., Calazans, A. T. S., Almeida, F. V., Costa, P. H. T., & Lima, F. (2020). Design thinking use in agile software projects: Software developers' perception. *ICEIS (2)*, 217–224.

Cerdeira, N., & Kotashev, K. (2021). Startup failure rate: Ultimate report + infographic [2021]. Retrieved February 3, 2022, from https://www.failory.com/blog/startup-failure-rate

Cohen, D., Lindvall, M., & Costa, P. (2004). An introduction to agile methods. *Adv. Comput.*, *62*(03), 1–66.

Dan, Y., & Chieh, H. C. (2008). A reflective review of disruptive innovation theory. *PICMET'08-2008 Portland International Conference on Management of Engineering & Technology*, 402–414.

Design Council. (2007). *Eleven lessons : Managing design in eleven global companies*. Retrieved March 3, 2022, from https://www.designcouncil.org.uk/our-work/skills-learning/resources/11-lessons-managing-design-global-brands

Design Council. (2019a). *The double diamond: A universally accepted depiction of the design process*. Retrieved March 4, 2022, from https://www.designcouncil.org.uk/news-opinion/double-diamond-universally-accepted-depiction-design-process

Design Council. (2019b). *What is the framework for innovation? design council's evolved double diamond*. Retrieved March 3, 2022, from https://www.designcouncil.org.uk/news-opinion/what-framework-innovation-design-councils-evolved-double-diamond

Dobrigkeit, F., de Paula, D. et al. (2017). The best of three worlds-the creation of innodev a software development approach that integrates design thinking, scrum and lean startup. *DS 87-8 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 8: Human Behaviour in Design, Vancouver, Canada, 21-25.08. 2017*, 319–328.

Dobrigkeit, F., de Paula, D., & Carroll, N. (2020). Innodev workshop: A one day introduction to combining design thinking, lean startup and agile software development. *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)*, 1–10.

Dobrigkeit, F., Paula, D. d., & Uflacker, M. (2019). Innodev: A software development methodology integrating design thinking, scrum and lean startup. *Design thinking research* (pp. 199–227). Springer.

Eisenmann, T. R., Ries, E., & Dillard, S. (2012). Hypothesis-driven entrepreneurship: The lean startup. *Harvard Business School Entrepreneurial Management Case*, (812-095).

Esportal. (2022). Esportal - homepage. Retrieved May 11, 2022, from https://esportal.com/no/

Felin, T., Gambardella, A., Stern, S., & Zenger, T. (2019). Lean startup and the business model: Experimentation revisited. *Forthcoming in Long Range Planning (Open Access).*

Fowler, M. (2019). Micro frontends. Retrieved May 19, 2022, from https://martinfowler.com/articles/micro-frontends.html

Fowler, M., Highsmith, J. et al. (2001). The agile manifesto. *Software development*, *9*(8), 28–35.

Frontendmasters. (2022). What is a front-end developer? Retrieved May 18, 2022, from https://frontendmasters.com/guides/front-end-handbook/2018/what-is-a-FD.html

Grant, R. M., & Baden-Fuller, C. (2004). A knowledge accessing theory of strategic alliances. *Journal of management studies*, *41*(1), 61–84.

Gray, R. (2021). 10 years since 'the lean startup': A product developer's perspective. Retrieved May 29, 2022, from https://www.forbes.com/sites/forbesbusinesscouncil/2021/02/17/10-years-since-the-lean-startup-a-product-developers-perspective/?sh=607ae72e7d8f

Guba, E. G., & Lincoln, Y. S. (1989). *Fourth generation evaluation*. Sage.

Gyenes, L. A. (1991). Build the foundation for a successful joint venture. *The Journal of Business Strategy*, *12*(6), 27.

Häger, F., Uflacker, M. et al. (2016). Time management practice in educational design thinking projects. *DS 85-2: Proceedings of NordDesign 2016, Volume 2, Trondheim, Norway, 10th-12th August 2016*, 319–328.

Highsmith, J. (2004). Extreme progamming. Retrieved May 13, 2022, from https://cmapspublic.ihmc.us/rid=1240967864977_1089895951_5483/Extreme%20Programming.pdf

Hoda, R., Salleh, N., & Grundy, J. (2018). The rise and evolution of agile software development. *IEEE software*, *35*(5), 58–63.

Hultberg, C. (2021). Technical decision-making in startups and its impact on growth and technical debt.

IBM. (2022). What is software development? Retrieved May 18, 2022, from https://www.ibm.com/topics/software-development

Iskander, N. (2018). Design thinking is fundamentally conservative and preserves the status quo. *Harvard Business Review*, *5*(09), 2018.

Jolak, R., Wortmann, A., Liebel, G., Umuhoza, E., & Chaudron, M. R. (2021). Design thinking and creativity of colocated versus globally distributed software developers. *Journal of Software: Evolution and Process*, e2377.

Jørgensen, M., & Sjøberg, D. I. (2004). The impact of customer expectation on software development effort estimates. *International Journal of Project Management*, *22*(4), 317–325.

Kahneman, D. (2011). *Thinking, fast and slow*. Macmillan.

Kahneman, D., & Tversky, A. (1977). *Intuitive prediction: Biases and corrective procedures* (tech. rep.). Decisions and Designs Inc Mclean Va.

Kenton, W. (2020). Sweat equity. Retrieved May 29, 2020, from https://www.investopedia.com/terms/s/sweatequity.asp

Killing, P. (2013). *Strategies for joint venture success (rle international business)*. Routledge.

Kniberg, H., & Skarin, M. (2010). *Kanban and scrum-making the most of both*. Lulu. com.

Knoema. (2022). Newzoo. Retrieved May 29, 2022, from https://knoema.com/atlas/sources/Newzoo

Larrick, R. P. (2004). Debiasing. *Blackwell handbook of judgment and decision making*, 316–338.

Liker, J. K., & Morgan, J. M. (2006). The toyota way in services: The case of lean product development. *Academy of management perspectives*, *20*(2), 5–20.

Lizarelli, F. L., Torres, A. F., Antony, J., Ribeiro, R., Salentijn, W., Fernandes, M. M., & Campos, A. T. (2021). Critical success factors and challenges for lean startup: A systematic literature review. *The TQM Journal*.

McGregor, L., & Doshi, N. (2015). How company culture shapes employee motivation. *Harvard Business Review*, *11*, 1–13.

Ministry of Culture. (2019). Spillerom - dataspillstrategi 2020–2022. *Ministry of Culture*, *V-1012 B*.

Molokken, K., & Jørgensen, M. (2003). A review of software surveys on software effort estimation. *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings.*, 223–230.

Mone, E., Eisinger, C., Guggenheim, K., Price, B., & Stine, C. (2011). Performance management at the wheel: Driving employee engagement in organizations. *Journal of Business and Psychology*, *26*(2), 205–212.

Müller, R. M., & Thoring, K. (2012). Design thinking vs. lean startup: A comparison of two user-driven innovation strategies. *Leading through design*, *151*, 91–106.

Newzoo. (2021). Newzoo. Retrieved May 11, 2022, from https://newzoohq.medium.com/the-games-market-and-beyond-in-2021-the-year-in-numbers-38206842216

Newzoo. (2022). Newzoo. Retrieved May 11, 2022, from https://newzoo.com

Nikitina, N., Kajko-Mattsson, M., & Stråle, M. (2012). From scrum to scrumban: A case study of a process transition. *2012 International Conference on Software and System Process (ICSSP)*, 140–149.

Piral. (2022). Getting started. Retrieved May 18, 2022, from https://docs.piral.io/guidelines/tutorials/02-getting-started

Pulse of The Profession. (2018). Success in disruptive times.

Rae, J. (2014). Design can drive exceptional returns for shareholders. *Harvard Business Review*.

Ries, E. (2011). *The lean startup*. New York: Crown Business.

Schillaci, C. E. (1987). Designing successful joint ventures. *The Journal of Business Strategy*, *8*(2), 59.

Schwaber, K., & Sutherland, J. (2011). The scrum guide. *Scrum Alliance*, *21*(1).

Shiotsu, Y. (2021). A beginner's guide to back-end development. Retrieved May 19, 2022, from https://www.upwork.com/resources/beginners-guide-back-end-development

Signoretti, I., Salerno, L., Marczak, S., & Bastos, R. (2020). Combining user-centered design and lean startup with agile software development: A case study of two agile teams. *International Conference on Agile Software Development*, 39–55.

Snihur, Y., Lamine, W., & Wright, M. (2021). Educating engineers to develop new business models: Exploiting entrepreneurial opportunities in technology-based firms. *Technological Forecasting and Social Change*, *164*, 119518.

Tracxn. (2021). Gtop esports league organizers startups. Retrieved May 11, 2022, from https://tracxn.com/d/trending-themes/Startups-in-Esports-League-Organizers

Tschimmel, K. (2012). Design thinking as an effective toolkit for innovation. *ISPIM Conference Proceedings*, 1.

Turk, D., France, R., & Rumpe, B. (2002). Limitations of agile software processes. *Third International Conference on eXtreme Programming and Agile Processes in Software Engineering (XP 2002)*, 43–46.

Turk, D., Robert, F., & Rumpe, B. (2005). Assumptions underlying agile software-development processes. *Journal of Database Management (JDM)*, *16*(4), 62–87.

Unyscape. (2019). Scrum vs waterfall – which one is more suitable for your project [2019]. Retrieved March 18, 2022, from https://unyscape.com/scrum-vs-waterfall-which-one-is-more-suitable-for-your-project/

Wijman, T. (2021). *Global games market to generate $175.8 billion in 2021; despite a slight decline, the market is on track to surpass $200 billion in 2023*. Retrieved May 11, 2022, from https://newzoo.com/insights/articles/global-games-market-to-generate-175-8-billionin-2021-despite-a-slight-decline-the-market-is-on-track-to-surpass-200-billion-in-2023/

Womack, J. P., & Jones, D. T. (1996). *Lean thinking—banish waste and create wealth in your corporation*. New York, NY: Simon; Schuster.

Womack, J. P., Jones, D. T., & Roos, D. (1990). *The machine that changed the world*. New York, NY: MacMillan Press.

Yang, C., Liu, C., & Su, Z. (2019). Research and application of micro frontends. *IOP conference series: materials science and engineering*, *490*(6), 062082.

Yin, R. K. (2011). *Qualitative research from start to finish*. Guilford publications.

Yordanova, Z. (2020). Lean startup and agile for addressing educational challenges. *2020 7th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE)*, 1–4.

York, J., & York, J. (2019). The limits to lean startup for opportunity identification and new venture creation. *Arch Bus Adm Manag*, *2*, 131.

Zorzetti, M., Signoretti, I., Salerno, L., Marczak, S., & Bastos, R. (2022). Improving agile software development using user-centered design and lean startup. *Information and Software Technology*, *141*, 106718.

Appendix A

Interview Guide

A.1 Introduction

1. Tell us about your position in ORG
    - Earlier work experience
2. Tell us abit about ORG
    - First meeting with ORG
    - The goal of ORG
    - The user group

A.2 Main section

Can you tell us about ORGs history from the start until now?

Guidance points/follow-up questions if the interviewee does not elaborate within certain topics:

- Was there a clear strategy? If yes, what kind? How did it go?
- Have there been major changes along the way?
- How is a typical work week structured?

- Why was it decided to create a joint venture and not a standard company with one owner?
- How was the situation affected by the fact that there were several owners?
- Tell us about the process of adapting to the target group and making the product user-friendly.
- Tell us about the development of the MVP
- What kind of work methodology has been used? How has it worked?
- Is the work methodology affected by the fact that it is a joint venture?
- Tell us about how feedback has been given and handled. Who do you get feedback from?
- You are not that many employees in ORG, how do you think the communication worked? Is there openness and acceptance for differing opinions? How does the communication affect the results?
- Have there been situations with longer waiting periods to wait for clarifications? Was it affected by the fact that it is a joint venture?
- Is there a common agreement on what the goal is? (Has it always been that?)
- Do you have any thoughts about ORG's financial situation?

A.3 Summary/Conclusion

1. What do you think about the future of ORG?
2. Based on what you mentioned, would you do something different if you were to start over? Work Methodology, Strategy, MVP etc ..
3. What are your thoughts on the fact that ORG is a joint venture?

   Give a brief summary of what has been said:
4. Is there anything you want to add?

Thank you very much for coming for an interview. If you think of something that could be relevant, dont hesitate to contact us.

Appendix B

Intervjuguide

B.1 Introduksjon

1. Fortell litt om din stilling i ORG.
   - Tidligere arbeidserfaring
2. Fortell litt om ORG.
   - Første møte med ORG
   - Målet med ORG
   - Brukergruppen

B.2 Hoveddel

Kan du fortelle om ORGs historie fra oppstartsfasen til nåværende tidspunkt?

Veiledningspunkter/oppfølgingsspørsmål dersom intervjuobjekt ikke utdyper innenfor visse temaer:

- Var det en tydelig strategi? Hvis ja, hva slags? Hvordan gikk det?
- Har det skjedd større forandringer underveis?
- Hvordan er en typisk arbeidsuke strukturert?

- Hvorfor ble valget en joint venture og ikke en heleid bedrift av enten *Eier1* eller *Eier2*? Evt frittstående selskap.
- Hvordan ble situasjonen påvirket av at det var flere eiere?
- Fortell om prosessen for å tilpasse seg målgruppen og gjøre produktet brukervennlig.
- Fortell om utviklingen en MVP'en
- Hva slags arbeidsmetodikk er blitt brukt? Hvordan har det funket?
- Blir arbeidsmetodikken påvirket av at det er et fellesforetak (Joint venture)?
- Fortell om hvordan tilbakemeldinger har blitt gitt og håndtert. Hvem får dere tilbakemelding fra?
- Dere er ikke så mange i ORG, hvordan syntes du kommunikasjonen har fungert? Er det åpenhet og aksept for meninger? Hvordan påvirker kommunikasjonen resultatet?
- Har det vært situasjoner med lengre venteperioder for å vente på avklaringer? Ble det påvirket av at det er et fellesforetak?
- Er det en felles enighet om hva som er målet? (Har det alltid vært det?)
- Har du noen tanker rundt ORGs økonomiske situasjon?

B.3 Oppsummering/Avslutning

1. Hva tror du om fremtiden til ORG?
2. Basert på det du har nevnt, ville du gjort noe annerledes om du skulle startet på nytt? Arbeidsmetodikk, Strategi, MVP osv..
3. Hva er dine tanker rundt det at ORG er et fellesforetak?

   Gi en kort oppsummering av det som er sagt:

4. Er det noe du vil legge til?

Tusen takk for at du ville stille til intervju. Dersom du kommer på noe senere du føler er relevant, ikke nøl med å ta kontakt.