NTNU
Norwegian University of
Science and Technology

Master's thesis

2020

Nicolai Cappelen Michelet

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Computer Science

Nicolai Cappelen Michelet

Software Engineering Practices and Launch Strategies in Startups Developing Multi-Sided Platforms

June 2020

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Software Engineering Practices and Launch Strategies in Startups Developing Multi-Sided Platforms

Nicolai Cappelen Michelet

Software Engineering
Submission date: June 2020
Supervisor: Trond Aalberg

Norwegian University of Science and Technology
Department of Computer Science

This thesis is dedicated to everyone who take a leap of courage to embark on an entrepreneurial journey

Abstract

In recent years there has been an increase in the focus on software development in startup companies (Unterkalmsteiner, 2014). However, the current literature is not enough to base an understanding of how software engineering practices could help software startups (Unterkalmsteiner, 2016). Agile methodologies have been considered as the most viable process since they embrace change, allowing software development to adapt to the business strategy. While the principles of lean and agile development are well known, how they are used in multi-sided platform (MSP) start-ups specifically is not. Besides, launch strategies of MSP's are rarely empirically examined (Schirrmacher, 2017).

The objective of this master thesis is *to understand the ongoing software development practices, engineering activities, and launch strategies in MSP startups. In particular, how platform startups support and understand their core transaction.* Data was gathered by conducting semi-structured interviews with MSP startups. The analytical frameworks 6 W's and 3 How's and Platform Business Model Canvas are utilized to analyze the startups.

The findings show that the MSP startups use various agile methodologies and requirement engineering practices. Further, the findings indicate that customer validation and requirement gathering should be emphasized more in the process of supporting the core transaction of the platform, especially for startups in an early stage. Startups in mature stages show more adeptness at applying agile methodologies than younger startups. Moreover, the results suggest that the methodologies and engineering practices used do not have any particular influence on the choice of launch strategy. Also, the MSP startups focus on the development of 'power features' but are generally unaware of their chosen launch strategy.

Sammendrag

De siste årene har det vært en økning i fokuset på programvareutvikling i oppstartsbedrifter (Unterkalmsteiner, 2014). Den nåværende litteraturen er ikke nok til å basere en forståelse av hvordan programvareutvikling kan hjelpe oppstartsbedrifter (Unterkalmsteiner, 2016). Smidig utviklingsmetodikk har blitt sett på som den mest attraktive prosessen siden den omfavner endringer slik at programvareutviklingen kan tilpasse seg forretningsstrategien. Hvordan prinsippene for smidigutvikling brukes i flersidig-plattformer (FSP) er ikke velrenommert. Dessuten blir lanseringsstrategier av FSPer sjeldent undersøkt empirisk (Schirrmacher, 2017).

Målet med denne masteroppgaven er *å forstå den pågående programvareutviklingsprosessen og lanseringsstrategiene i FSP oppstartsbedrifter. Det undersøkes hvordan disse plattforms bedriftene støtter opp under og forstår sin kjernetransaksjon.* Data ble samlet inn ved å gjennomføre semi-strukturerte intervjuer med FSP oppstartsbedrifter. De analytiske rammeverkene 6 W og 3 How's og Platform Business Model Canvas ble brukt til å analysere bedriftene.

Funnene viser at FSP oppstartsbedrifter bruker forskjellige smidige arbeidsmetoder og metoder for programvarespesifikasjon. Videre indikerer funnene at kundevalidering og kravinnsamling bør vektlegges mer for å kunne støtte opp under kjernetransaksjonen til plattformen, spesielt for oppstartsbedrifter i en tidlig fase. Oppstartsbedrifter i mer modne stadier viser en større grad av bruk av smidige arbeidsmetoder enn det yngre oppstartsbedrifter gjør. Resultatene antyder dessuten at valg av arbeidsmetodologier og programvarespesifisering som brukes ikke har noen særlig innflytelse på valget av lanseringsstrategi. FSP oppstartsbedrifter fokuserer på utvikling av 'spesielle funksjoner', men er generelt ikke så klar over de ulike lanseringsstrategiene som finnes.

Preface

This thesis is part of the Masters program in Informatics at the Norwegian University of Science and Technology (NTNU), Faculty of Information Technology and Electrical Engineering, Department of Computer Science. It was written during the fall of 2019 and spring of 2020.

I am grateful to have such freedom with regards to choice of topic and being able to write about something that I am truly passionate about. Due to the world-wide COVID-19 pandemic, many people have lost their jobs and the impact may continue to affect the whole world for years to come. Startups are an exciting, yet grueling endeavour and in these strange times there are unfortunately much higher uncertainties than before. We are only in the beginning of an unprecedented shift to a platform economy and so a need for this type of research is clear, especially now with the consensus that the digital transformation has leapt years ahead of the 'predicted schedule'. By conducting research on MSP startups' development, core transactions, and launch strategies I hope to provide some useful insights for software startup researchers and practitioners alike and aid them in the relentless pursuit of their vision.

The support and feedback I have received from friends and family reminds me of the profound gratitude I feel for having them in my life and the opportunity given to me to be able to complete higher education. Thank you Dad for always stimulating my curiosity and teaching me the importance of perseverance. Thank you Mom for showing me what true compassion and unselfishness is all about. Thank you to my brothers for always challenging me on every level.

I would like to thank the nine companies that allowed themselves to be interviewed in relation to my thesis. I also want to extend my sincere gratitude to my supervisor Trond Aalberg and external supervisor Anh Nguyen for support and valuable guidance along the way. Without them this thesis would not be possible.

Oslo, June 2, 2020
Nicolai Cappelen Michelet

Table of Contents

Abstract i

Sammendrag i

Preface ii

Table of Contents iv

List of Tables v

List of Figures vii

Abbreviations viii

1 Introduction 1

1.1 Motivation 1

1.2 Research Questions 2

1.3 Research Scope 3

1.4 Research Process 4

1.5 Thesis Outline 4

2 Background 7

2.1 Software Startups 9

2.1.1 Lean Startup 9

2.1.2 Agile Software Development 10

2.2 Software Startup Engineering 11

2.2.1 Requirement Engineering 12

2.2.2 Technical Debt 13

2.3 Multi-Sided Platforms 13

2.3.1 Core Transaction 14

2.3.2 Network effects 15

2.3.3 Network Management 16

2.4 Launch Strategies 17
2.4.1 Types of Launch Strategies 17
2.4.2 Sequential 18
2.4.3 Simultaneous 19

3 Methodology 21
3.1 Research Method 21
3.2 Research Questions 22
3.3 Case and Subject Selections 23
3.3.1 Interview Process 23
3.4 Data Collection 24
3.5 Data Analysis 25
3.6 6W3H Analytical Framework 25
3.6.1 Descriptions of the 6W3H Framework 26
3.7 Platform Business Canvas Model 27

4 Results 29
4.1 Case Descriptions 29
4.2 Platform Development 34
4.2.1 Which software methodologies are used in MSP startups? 34
4.2.2 Which requirement engineering practices are used in MSP startups? 38
4.3 Platform Launch 45
4.3.1 Piggybacking 46
4.3.2 Subsidy 47
4.3.3 Single Target Group 47
4.3.4 Platform Staging 48
4.3.5 Big-bang 48
4.4 Platform Development and Launch 49
4.5 Discussion 52
4.6 Validity of Findings 53

5 Conclusion 55
5.1 Summary 55
5.2 Future work 56
5.3 Research process experiences 56

Bibliography 57

Appendix 63

List of Tables

2.1 Launch strategies and the order of entry 19

3.1 Interviews 24

4.1 Case descriptions 29
4.2 S1 core transaction 30
4.3 S2 core transaction 30
4.4 S3 core transaction 31
4.5 S4 core transaction 31
4.6 S5 core transaction 32
4.7 S6 core transaction 32
4.8 S7 core transaction 33
4.9 S8 core transaction 33
4.10 S9 core transaction 33
4.11 Overview over startup methodologies 37
4.12 Overview over chosen launch strategies 49

List of Figures

2.1 Relationships between background material 8
2.2 Build Measure Learn Cycle 10
2.3 Airbnb's user growth over time 16

3.1 Research process 22
3.2 6W3H Framework 26
3.3 Platform business canvas model 28

5.1 Consent form 64
5.2 Interview Guide in English 65
5.3 Interview Guide in Norwegian 66
5.4 General mail to companies 67

Terms and Abbreviations

MSP	=	Multi-sided Platform
IT	=	Information Technology
SE	=	Software Engineering
RE	=	Requirement Engineering
TD	=	Technical Debt
MVP	=	Minimum Viable Product
SWEBOK	=	Software Engineering Body of Knowledge
IEEE	=	Institute of Electrical and Electronics Engineers
API	=	Application Programming Interface
SDK	=	Software Development Kit
CTO	=	Chief Technical Officer
6W3H	=	Six W and Three H framework
BMC	=	Business Model Canvas
PBMC	=	Platform Business Model Canvas
WOM	=	Word of Mouth effect

Chapter 1

Introduction

This introduction aims to describe the irrefutable importance platforms have in our world and set the stage for the whole thesis. The remainder of the chapter proceeds as follows: Section 1.1 presents the motivation for this research. Section 1.2 presents the research questions. Section 1.3 defines the scope of the research. Section 1.4 explains the chosen research method and process. Section 1.5 outlines the rest of the thesis.

1.1 Motivation

Marketplaces are places where people transact. We have seen the emergence of powerful corporations that enable new types of markets. Companies such as Amazon, Facebook, Google, Salesforce, and Etsy are creating online structures that enable a large range of human activities (Kenney, 2016). Payment platforms like Apple Pay, Square, and PayPal are disrupting the financial industry while peer-to-peer platforms such as Airbnb and Uber are changing the way people work. We see that platforms pave the way for radical changes in how we socialize and create value in the economy.

Information technology (IT) tremendously increases the opportunities for building powerful platforms and enables multi-sided platforms (MSPs) to have a wide-ranging impact on businesses, workers, and consumers (Hagiu, 2009). Salminen (2014) defines multi-sided platforms as “places of interaction”, in which technology is employed to connect different user groups. Geographical boundaries are drastically erased due to the inherent characteristics of the Internet, enabling practically anyone to build new innovative MSPs.

Over the past few years, software startups have garnered increased research interest in the Software Engineering (SE) community (Unterkalmsteiner, 2016). Software startups face unique challenges when it comes to starting their business. This includes limited time, resources, and extreme uncertainty (Unterkalmsteiner, 2014). Software engineering in startups is a relatively new field of research but has revealed that startups tend to use agile and lean methods in an ad-hoc manner.

Multi-sided platforms startups face a coordination problem. Without a network of users, MSPs remain valueless. This is sometimes called the ”chicken and egg problem”. This occurs in new platforms when they do not initially create enough value to attract new users. The value of the product is often derived directly from the interactions that take place on the platform. For example, every new listing on Airbnb creates value for people searching for places to stay and that attracts even more tenants, thereby increasing the value for landlords. The development of the product is therefore of utmost importance to create a frictionless experience for the user that best facilitates interactions. This added element can explain why MSP startups are especially prone to failure. A platform can only create value if users interact with each other and so we will see new MSPs struggle to solve the chicken-and-egg dilemma (Stummer, 2018). Building a platform creates unique challenges for practitioners due to the complex nature of software-based products, the vast array of engineering practices, and the chicken and egg dilemma.

Therefore, more research should be provided to both support software development and launch strategies in the unique context of MSPs. In the absence of research on multi-sided platform startups, this thesis aims to be a first step in *investigating the requirement engineering practices and launch strategies used by such companies.* The end-user could care less about how MSPs are developed, but for software engineers, entrepreneurs, researchers, and others who aim to build and understand platforms, it is imperative to be aware of the software development that enables the core transaction. Getting this transaction right is crucial for any platform company (Moazed, 2016). Particularly for startups as poor initial prioritization of requirements can doom the venture.

1.2 Research Questions

Inadequacies in applying engineering practices could be a significant contributing factor to startup failure (Klotins E., 2015). Little is known as to how multi-sided platforms perform their software development and launch. More specifically, which methodologies and engineering practices they apply to support the core transaction. This thesis aims to investigate this topic through the following research questions:

1. How do multi-sided platform startups develop their platform in order to support their core transaction?
    (a) Which software methodologies are used in MSP startups?
    (b) Which requirement engineering practices are used in MSP startups?
2. Which launch strategies do MSP startups use?
3. How do MSP startups coordinate the development of the platform with launch?

The first research question has two sub-questions that aim to investigate how MSP startups develop their platform. As this thesis focuses on the core transaction of platforms, a critical part is figuring out what to develop. Requirement engineering was identified as the most relevant software engineering principle because it explicitly targets the question of what to build. This is explained at the beginning of chapter 2. The second question is geared towards the unique challenge platform startups face and investigates which launch strategies are used. The third question attempts to bridge the two questions and explore if any relationship(s) exist between software development and the launch of the platform.

1.3 Research Scope

The topic of this thesis is both how MSP startups develop their software and how they launch. The focus is on what is called the ’core transaction’ of MSP startups and how software methodologies can help in a startup’s life cycle. More specifically, the processes of customer validation, requirement prioritization, and measurement. To examine every aspect of the software engineering process in MSP startups was considered too big of a scope. Figuring out what to build is crucial for MSPs and so requirement engineering was identified as especially relevant to investigate.

In addition, this thesis explores the relationship between launch strategies and platform development. The applicable research area is within software startup engineering as well as management and business development research. Even though management, finance, and human resources are important areas of study in startups, they are not particularly considered in this thesis. It should be noted that the studied MSPs are *online platforms* (explained further in section 2.3).

This thesis studied 9 multi-sided platform startups. Eight startups are based in Norway, while one is based in Finland. The industries and markets vary in which these startups operate. There was no focus on which type of industry as it was not deemed relevant to the overall research questions. The platform companies were selected based on the following list of inclusion criteria:

- A *multi-sided online platform* that facilitates some sort of transaction between two or more sides.
- The company classifies itself as a startup
- Startup was located in the Nordic region

.
The selection process is explained further in Section 3.3. This thesis studied Nordic startups and the results may therefore not be transferable to multi-sided platforms outside of this region. However, this is just a stated formality because MSP’s are global ventures (depending on the core transaction). In this light, the results can be seen as applicable for every multi-sided platform startup as defined in this thesis.

1.4 Research Process

A multiple case study is considered suitable for software engineering research. To collect data, semi-structured interviews were conducted. A consent form was issued to the interviewees, and all interviews took place virtually, over video meetings when possible. The interviews were then translated, if necessary, and transcribed. This was an ongoing process, thus questions were often updated after each interview when new knowledge was acquired. Interviewees were selected based on having some technical knowledge or insight into the platform development. The selected persons' roles ranged from founders and CTO's, to developers. Data was collected from relevant startup websites, background research, and in-depth semi-structured interviews. The data from the interviews was then coded into categories using a thematic analysis process. These categories were used as a basis for structuring the results and in turn answering the research questions. The validity of the research is examined in section 4.6. The whole research process is described in detail in chapter 3.

To explain how the startups were analyzed and how the core transaction was identified, two frameworks are utilized. The first is the 6W3H framework from Nguyen-Duc (2020). This is a general analytical framework that provides the foundation for contextualizing MSP startups. To hone in on the core transaction, the Platform Business Model Canvas (PBMC) was identified as particularly relevant. This canvas is used to identify and understand the core transaction of MSP startups. The frameworks are explained in section 3.6.1.

1.5 Thesis Outline

This thesis contains 5 chapters (including this one) and has the following outline:

Chapter 2 dives into the various literature on software startups, multi-sided platforms, and software engineering. Also, it explains the background of launch strategies for MSPs. This sets the stage for investigating both software development practices and launch strategies in chapter 4.

Chapter 3 explains the chosen methodology including case selections, data collection and analysis, and chosen frameworks. By rooting the research in well-known practices and methods this thesis aims to contribute to the existing research body on software engineering startups.

Chapter 4 discusses the results and validity of the research. First, the MSP startups are described and their core transaction examined by means of the PBMC. Then the results are presented in conjunction with a discussion. The chapter concludes with a further general discussion and validity of the research.

Chapter 5 concludes the thesis by summarizing the chosen research areas, the research process, and the key findings of the study. In addition, key project developments and future research areas are discussed.

Appendices include:

- **A** The consent form sent to interviewees
- **B** Interview Guide in Norwegian and English
- **C** General email templates sent to startups

Chapter 2

Background

This thesis is based on various literature on software startups, multi-sided platforms, requirement engineering, and launch strategies.

Multi-sided platform startups are viewed in this thesis as a subset of software startups. Regardless of the categorization software startups are assigned, the company needs to develop software. This leads us to the field of software startup engineering. Needless to say, this is a subset of software engineering. Software engineering in startups represents a segment that has mostly been neglected in research studies (Unterkalmsteiner, 2016). Processes are tailor-made and have been proven to be done ad-hoc. In current software startups, there has been a wide acknowledgment of the benefits that agile software development processes can have on projects. Principles from the Lean Startup movement permeate much of these processes, also known as methodologies. Methodologies detail work-flow and ”enable software developers to produce higher quality software in a shorter period of time” (Livermore, 2008). There is no full explanation in this thesis of various software methodologies as the focus is more on which practices are used. Also, the hypothesis is that MSP startups use agile practices ad-hoc. However to provide some context Scrum and Kanban were identified early on as methodologies that were interesting to research and as a result are included briefly in this background.

To easily identify the different topics and their relationship with each other figure 2.1 is presented below.

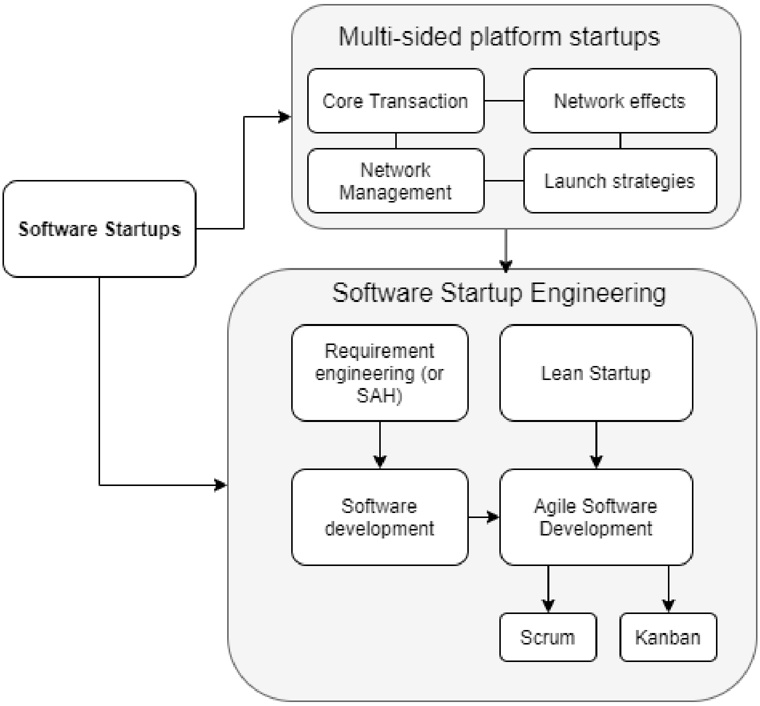


Figure 2.1: Relationships between background material

The background on software engineering in this thesis stems from the Software Engineering Body of Knowledge (SWEBOK) (Bourque, 2014). During the past fifty years or so software engineering has grown from a conference catchphrase into an engineering profession. The IEEE (Institute of Electrical and Electronics Engineers) are the authors of SWEBOK version 3. It is the most up-to-date source on the body of knowledge in software engineering. It is far out of scope to be able to encompass the whole field of software engineering in a master thesis. Rather, as this thesis focuses on the *core transaction of multi-sided platform startups* requirement engineering was identified as the most relevant topic for further study.

The rest of the chapter proceeds as follows: Section 2.1 defines software startups and explains the development practices that stem from Lean startup. Section 2.2 dives into the world of software startup engineering with a focus on requirements. Section 2.3 explains the background around multi-sided platforms with a focus on the core transaction and network management. Section 2.4 explains the coordination problem faced by startups and possible launch strategies that can be implemented.

2.1 Software Startups

Software startups are increasingly important in generating innovative products and services that impact the global economy (Nguyen-Duc and Abrahamsson, 2015). A startup is a human institution designed to create a new product or service under conditions of extreme uncertainty (Ries, 2011). A startup can also be described as a temporary organization that creates high-tech innovative products and as being on the search for a scalable business model (Blank, 2012). The term software startup was first used in 1994 by (Carmel, 1994), who argued that software is becoming a product on its own. However, there is no consensus on the definition, despite the general understanding that software startups have certain traits. These traits include rapid growth, goals for scalability, extreme uncertainty, and innovative products.

The challenges startups face in contrast to traditional organizations are well-known. Software startups share common characteristics with other types of startups (for example resource scarcity). However, software startups face a ”...wave of technological change frequently happening in the software industry, such as new computing and network technologies, and an increasing variety of computing devices” (Unterkalmsteiner, 2016). Unterkalmsteiner (2014) highlights challenges such as dependency on third-parties, time-pressure, small teams with limited knowledge, low experience, uncertainty, and lack of resources, among others. Given all the challenges, it is therefore not a surprise that the majority of software startups fail within two years of their creation. However, it is primarily due to self-destruction rather than competition (Crowne, 2002). Due to this, it is worth exploring how startups work and what factors lead to their success.

As customers and products often are unknown, the success of startups depends on how fast they can prototype to test business ideas (Sutton, 2000). Startups tend not to follow established process frameworks, leaving more liberty and allowing for customized and dynamic work methods in its ways of working (Berg, 2018).

2.1.1 Lean Startup

The term ’lean startup’ has flourished in recent years and is used in different contexts. Ries (2011) presented the Lean Startup method in 2008, based on lean principles first introduced by Toyota (Womack, 1990). Lean startup is an important methodology in understanding what product to develop. Startups tend to prefer time and cost over product quality (Yau, 2013) neglecting traditional process activities like formal project management, documentation, and testing (Unterkalmsteiner, 2014). The method provides principles for how to run a new business, where the objective is to grow the business as fast as possible. The method aims managing and creating startups, and to deliver services or products to customers. By iteratively turning ideas into products, measure customers’ satisfiability, and learn from their feedback, startups can accelerate their business. This process is referred to as the build-measure-learn (BML) feedback loop.

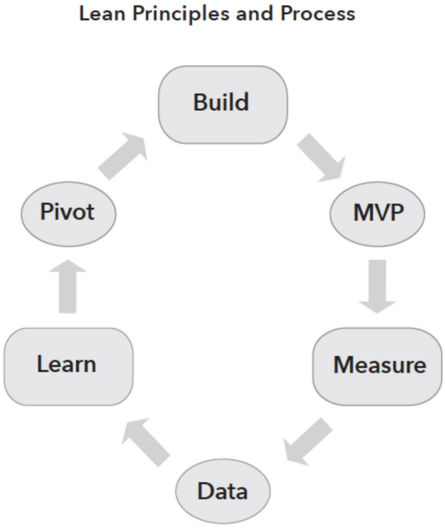


Figure 2.2: Build Measure Learn Cycle

Ries (2011) recommends testing the riskiest elements of a startup's plan first. To test fundamental business hypotheses, the recommendation is to test new versions of minimum viable products throughout product development. The concept of a minimum viable product (MVP) is important in Lean Startup. MVP's are developed to test the hypothesis as fast as possible and can take many forms. MVPs can range from post-it notes, varying degrees of finished prototypes, to completed software. Together with "startup," MVP is one of the most overused and misunderstood terms among practitioners (Nguyen-Duc, 2020).

The final step of the loop is whether to pivot or persevere. Based on the results of testing the MVP or prototype, ideally, certain conclusions will arise. The startup will then need to decide its plan of action. Pivots are an incredibly important step in startups. It is the strategic process of changing the course of action in order to "... test a fundamental hypothesis about a product, business model, or engine of growth (Ries, 2011). Given the nature of software startups, pivots are known to be key to the initial survival of the business (Bajwa, 2016). Famous examples of pivots include Twitter which originally started as a podcast service and Instagram which was a social check-in application hybrid with a game called Mafia wars (Nazar, 2013). Negative customer reaction is the major triggering factor contributing to pivots in startups (Bajwa, 2016). It is therefore not surprising that the most occurring type of pivot is the customer need pivot.

2.1.2 Agile Software Development

As IT emerged and became an established engineering field, there arose a need for a plan-driven approach. The first such example is the Waterfall method. This method is often cited as first proposed by Royce (1970) and divides software development into different phases and progress is seen as flowing steadily downwards (Royce, 1970). The model allows feedback loops and iterations but is today looked at as outdated because it offers low flexibility in the project cycle.

The term agile process arrived and was first used for agile manufacturing. Later Aoyama used the term for the first time in the context of software development (Aoyama, 1998). The term started to resonate in the developer community as agile methods seek to avoid unnecessary development that adds little value to the customer. The software development process is agile when ”...software development is incremental (small software releases, with rapid cycles), cooperative (customer and developers working constantly together with close communication), straightforward (the method itself is easy to learn and to modify, well documented), and adaptive (able to make last moment changes)” (Salo, 2017). Agile software development can be seen as a way to implement or operationalize some of the key values that permeate lean startup.

Many different agile methods are in use today (Adlin, 2010). XP (eXtreme Programming) is a set of well-known engineering techniques practices (Bosch, 2013). Scrum is another method in use and is used to manage software development. At the core of Scrum lies the idea of having sprints, planning games, daily stand-ups, and sprint reviews. There is also to be a dedicated Scrum master that is in charge of leading the practices. Scrum can be seen as a wrapper for the practices outlined in XP (Bosch, 2013). Another prominent agile methodology is Kanban. The main idea ”...is to accurately state what work needs to be done, and when it needs to be done” (Lei, 2017). By prioritizing tasks and defining workflow Kanban is a software methodology that practices the lean principle of ”Just-in-time” delivery. Also, Kanban preaches that developers do the right work at the right time given their skill sets. As such it is chosen here as a methodology that is particularly interesting for software startups because it is proven to reduce waste in software engineering projects (Anderson, 2010).

Using software development methodologies is not easy in software startups. Attempts to tailor lightweight processes to startups report failures: “Everyone is busy, and software engineering practices are often one of the first places developers cut corners” (Martin, 2007). Rejecting the notion of controlled and repeatable processes, startups prominently take advantage of reactive and low-precision engineering practices (Tanabian, 2005).

2.2 Software Startup Engineering

In the last few years, the interest in research on software startups has gained increased interest in the Software Engineering (SE) community, highlighted by the increased publication frequency (Unterkalmsteiner, 2016). Software startup engineering is the first set of concepts, terms, and activities for the software startup phenomenon, defined as “the use of scientific, engineering, managerial, and systematic approaches with the aim of successfully developing software systems in startup companies”.

Agile methods have proven to be able to build successful businesses. Perhaps because new requirements, features, or bugs are discovered faster with shorter development iterations. Following the build-measure-learn model gives a cycle-like form to the SE process in software startups. Nevertheless, startups very often develop something and then realize that users do not want it, even though validating assumptions as soon as possible is present in well-known startup development methodologies like Lean Startup and Customer Development. Interviewees still aware of these methodologies made these mistakes. In software engineering, there is an increasing amount of experience reports on insufficient supports for startup engineering and development (Nguyen-Duc and Abrahamsson, 2015).

Agile methods focus mostly on ’how’ to build software, but not on ’what’ to build. Focusing on technological solutions will not guarantee survival and success. To develop something valuable for customers, startups need to understand their real problems. Not understanding the issue at hand can be a potential root cause for other key challenges including acquiring the first paying customers. That leads us to the field of requirement engineering.

2.2.1 Requirement Engineering

Requirements are attributes that we discover before building products.

> *”It is a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.”* (Pandey, 2010).

Requirement engineering (RE) is a systematic approach in which the software engineer or team collects requirements from different sources and subsequently implements them into the development process. In software engineering, using unreliable RE techniques can cause software projects to fail (Tripathia, 2018). Due to the many challenges software startups face, RE processes are often basic (Klotins E., 2015) This often results in startups making products that are unsuitable for their target markets (Giardino, 2014). Morthen (2019) researched a software startup and the requirement engineering process. His research confirms that RE is ”...done ad-hoc, processes are tried out and either rejected or accepted and change as the context changes” (Morthen, 2019). Further, he argues that requirement engineering is not a suitable term for software startup engineering. He proposes Software Activity Handling as a more proper term as RE is a process in startups and not a set of static activities.

In startups, the customer is sometimes well-known and the software is then developed accordingly. When the customer defines the requirements, RE can be characterized as customer-driven. However, in startups, this is rarely the case because the market is not well-tested or known. When the customer is not defined, RE is classified as market-driven. Market-driven software development is where specific requirement elicitation techniques (prototyping) and time-to-market are key objectives (Rafiq, 2017). In a market-driven context, requirements tend to be invented by the software company, rarely documented, and validated only after the product is released in the market (Dahlstedt, 2003). Products that don’t meet customer needs are therefore common, resulting in the failure of new product releases (Alves, 2006).

In addition, startups must know what features to prioritize. Requirements prioritization defines what the team is focusing on and what they are building. In terms of planned releases, Klotins mentions two main approaches. Smaller releases allow for more frequent feedback and relate to continuous requirement validation. On the other hand, delivering a full product can lead to less overhead (only one release) and companies want to keep and attract customers by having a more polished product. However, companies run the risk of having developed irrelevant features and wasting precious time, money, and effort (Klotins, 2019).

2.2.2 Technical Debt

In software projects, there have been studies on the concept of technical debt (TD). TD is facing the challenge of accepting compromises in a system in one aspect in order to meet an urgent demand (Apa, 2020). It relates to the technical decisions made in software development that can bring benefits in the short term but may pose risks in the long term. For example, by making it difficult to implement certain features due to architectural decisions early one. Apa (2020) looked at how software startups in Uruguay perceive and manage TD. They found no unanimity on how startup practitioners perceive or manage TD. Those participants who declared that they try to manage technical debt belonged to the more prominent organizations and were linked to having the highest level of experience.

In general, research on how software startups develop their product is not comprehensive enough in depicting the picture to support them. Multi-sided platform startups also face the same issues as other software startups, yet little research is conducted on their development.

2.3 Multi-Sided Platforms

This section aims to explain the intricacies of how the term platform is defined. There are several definitions of the term in existing research and the way it is defined often depends on the field of study. Defining what is meant by the term platform is vital to avoid confusion because it is used differently in a wide variety of disciplines. In engineering, it is often used to describe systems or products that are physical. The product is inherently modular in behavior, allowing several products to be built on top of existing ones, thereby creating a product platform. The term is also used interchangeably often in Software-as-a-Service companies (as a platform of related software products). Ghazawneh (2015) builds on Tiwana (2010) definition of platform by defining digital platforms as: ”software-based external platforms consisting of the extensible code-base of a software-based system that provides core functionality shared by the modules that interoperate with it and the interfaces through which they interoperate”. This definition is only partly adopted in this thesis because many multi-sided platform startups do not have an application programming interface (API), or the ability for third-parties to help with value creation. A platform in this thesis is therefore not limited by this definition, because a marketplace can exist regardless of the architecture that allows for external interfaces (third-party developers). In this thesis a platform is simply defined as:

> *”...a business model that facilitates the exchange of value between two or more user groups, a consumer and a producer” (Moazed, 2016).*

This rules out a platform as just a piece of technology, or as an integrated suite of software products.

Multi-sided platform startups are a specific type of platform that creates value by enabling multiple user groups to interact (Rochet, 2006a). Included in the definition of multi-sided is also the often-used term two-sided platforms. Two-sided platforms are similarly defined as commercial digital networks serving two-sided markets where the value for one side depends on the number of participants on the other side (Schiff, 2003). Such multi-sided platforms may create value through ‘matchmaking’, i.e. reducing search and transaction costs between two user groups (Evans and Schmalensee, 2016). MSP startups are a subset of software startups that focus on marketplaces and connecting consumers and producers. However, the term MSP startup in itself is not well discussed in the literature.

Wertz (2015) introduced three phases two-sided platform startups can be in - seeding, growing, and scaling. The phases are defined by the number of users.

1. Seeding - platform owners invite and manually onboard users
2. Growing - both users groups are growing towards critical mass
3. Scaling - platform scales by itself

These classifications are used in this thesis on the studied platforms. Next, it becomes crucial to understand a platform’s core transaction and the network that it is trying to build.

2.3.1 Core Transaction

Every digital platform has a core transaction - a set of actions consumers and producers must complete in order to exchange value (Moazed, 2016). A core transaction can also be defined as a recurring transaction that creates value for the platform. This is reiterated with the following: ”... creating and capturing value is the ’core interaction’ of platforms” (Van Alstyne, 2016). The key to a platform’s success can be explained with repeatable and sustainable interactions (Choudary, 2015). Therefore the matchmaking process is perhaps the most important feature of the platform. By removing friction in the process, it allows for more of the interaction to occur. However, defining the exact features of the platform that aid in matchmaking is more challenging.

Another aspect to consider is here is the integration with third-parties. The biggest profits are gained when platforms are opened to third parties (Ailisto, 2016). This is often called boundary resources, platform transparency, or openness. It could refer to documentation, the availability of software development kits (SDKs) or APIs. For instance, by providing an API Facebook opened up for third-party developers and created the vast sub-market of applications that has over 150 million users every month (Nazir, 2009). Boundary resources can aid in creating value for platforms but are not viewed in this thesis as being part of the core transaction.

Korhonen (2017) analyzed 29 platform startups and how they support the core transaction, attract users, and provide features to their users. Of the studied startups, they find that most of them support the core transaction by ”providing an easier and unified interface for these services.” (Korhonen, 2017). The platforms provided communication, information exchange, and brought together the right kind of users with the right producer. Korhonen uses the platform canvas to illustrate the core transaction of the startups. The canvas emphasizes the central role of core interaction towards value capturing. They highlight four key elements of the platform canvas model that are especially important for this:

1. Users (or consumers)
2. Producers
3. Value proposition
4. Value capture

The users and producers are the central participants in the market. The platform provides some type of value to each side. The value proposition “describes the benefits customers can expect from your products and services” (Osterwalder, 2014). Most often it is reducing the time and cost of the transaction in question. For MSP startups it is crucial to focus on the interaction to match the most compatible users with each other and facilitate value co-creation. Value capture concerns how the platform can generate revenue thereby sustaining itself. That is not the main focus in this thesis, but it should be mentioned nonetheless as MSP businesses need to be profitable.

In contrast to general software startups, MSP’s often face the challenge of not providing any inherent value to users right off the bat. This gives rise to the chicken and egg problem. Some multi-side platforms provide one side of users with value from the start so they are not as reliant on acquiring the other side. Network effects and management is explained in the subsequent sections.

2.3.2 Network effects

A network in the presence of positive network effects can be viewed as a magnet - it repels users at first and then flips polarity and attracts. This happens when the value is greater than the cost of joining the platform. The typical growth of successful platforms can be illustrated in a hockey-stick-growth chart.

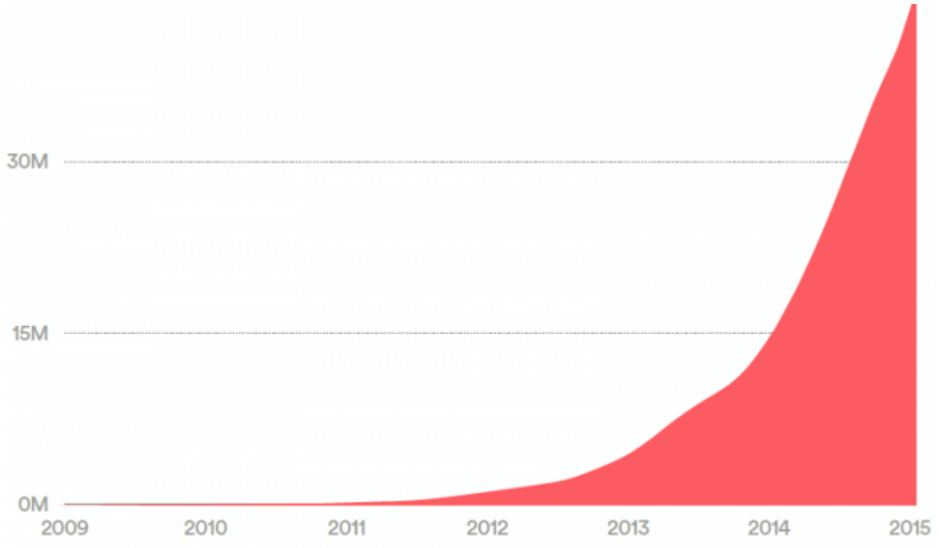


Figure 2.3: Airbnb's user growth over time

Network effects are present if users care about the participation and usage decisions of other users (Belleflamme, 2016). There are two types of network effects.

1. Direct (same-sided) - when the platform becomes more valuable if users in the same user group join.

2. Indirect (cross-sided) - when the value of the platform depends on users in different groups.

Network effects "...may be positive or negative depending on the circumstances" (Belleflamme, 2016). MSP's are reliant on network effects to grow, particularly indirect network effects. Sellers on eBay gain more value when they have access to a large number of buyers and vice versa.

2.3.3 Network Management

How a platform manages its network is crucial for its success. It would be easy if everyone could see the value and agree to join the network at the same time. In reality, coordination problems must be solved by incentivizing users to join. There are three main ways: (Moazed, 2016).

1. Monetary subsidies - giving money directly to consumers or producers so that they are incentivized to join the platform. This can also be done by referral fees. Uber for example, offered a free ride to users who successfully obtained another user. Food services like Foodora and Wolt offered money off the next order if a user referred a friend. One can also lower the price for a specific side of the platform.

2. Product features - creating special functionality for power users to increase loyalty and usage among this prized group. Verified users can get improved security and identity protection or better customer service. Instagram started as a great photo-taking app and provided a new feature - filters. This attracted many producers and provided content on the platform. Later it got more consumers and built a content platform around photos.

3. User sequencing - deliberately prioritizing certain user groups that others will want to interact with. For Twitter, this meant recruiting celebrities to the platform because many other users would join as a result. Twitter had designated VIP recruiters who specifically targeted celebrities.

The coordination problem is a unique challenge faced by MSP startups and arises in the presence of indirect network effects. The ‘chicken-and-egg’ problem implies that a sufficiently large number of users need to join for the platform to become valuable (Evans, 2010). This problem is so prevalent that several fields study it, among them are typically academics from information systems and economics (Stummer, 2018). It is believed that by combining ways to subsidize value it can allow a platform to overcome the coordination problem. However, choosing a launch strategy is difficult.

2.4 Launch Strategies

In platforms, competition no longer revolves around how to control the value chain, but around attracting generative activities associated with a platform. Platforms encompass a winner takes all mentality with often a major player becoming the prevalent platform in the market. Smaller platforms and startups can struggle in such scenarios.

The empirical evidence on launch strategies is limited and few are supported by real-life cases. There exist proposed launch strategies for platforms but they are rarely empirically examined. Also, the effectiveness of *combinations* of launch strategies is not well-researched (Stummer, 2018). Finding harmony between them all requires finesse and can be the reason why many platforms fail in an early stage. In addition, startup companies do not usually have the financial resources or time to solve the coordination problem through a massive marketing campaign, nor do they have the reputation of an established brand.

2.4.1 Types of Launch Strategies

There are several ways to deal with coordination problems, depending on the type of industry and platform business in question. Several general launch strategies for multi-sided platforms exist (Stummer, 2018). In addition, several ’sub-strategies’ fit into larger strategy themes which are outlined below. Those that are relevant for this thesis are:

1. *Single target group* focuses on one particular target group or market segment. There exist at least two variations of this strategy. One is focusing on important marquee users that will generate high value to the network and attract other users to join. These users can be opinion leaders, bring high-quality transactions to the MSP

(Binken, 2009) or serve as very active users (Wilson, 2009). The other variation is focusing on loyal users. There are various reasons why loyal users stick to a platform: lower price sensitivity (Rochet, 2003) higher sunk costs, or positive expectations for platform development (Zhu, 2012). For example, a rating system can help in retaining users because a user would have to rebuild their reputation from scratch if they switch to another similar platform.

2. *Platform staging* is a strategy where there is a distinct first stage that can help MSP's focus on one market at a time. It involves providing some immediate value to one of the sides. By providing a supply-side with fist-party content, it gives inherent value to users right off the bat. The OpenTable restaurant reservation service, for example, distributed booking management systems in the first stage. Restaurants used the system as a standalone application to manage table bookings. By having a large number of restaurants, OpenTable then opened up for users to book tables through their system and took a commission for every refereed booking their restaurants got.

3. Most literature initially focused on *subsidizing strategies* (Rochet, 2006b). It is a common strategy where MSP's typically offer free use of the platform for one 'subsidy' side while charging the 'money' side for participation or transactions. (Eisenmann, 2006). This can take many forms, such as free usage, price cuts, value-added services and even paying customers to use the platform. Uber implemented dynamic pricing. It meant raising prices when the supply of drivers was low. This increased drivers in the given areas and at the same time lowered the demand for the Uber service.

4. *Platform envelopment* is a strategy that focuses on addressing users that are already on an existing platform and directing them towards the desired platform. This means that MSPs can leverage existing relationships by for example integrating features on a target platform (Eisenmann, 2011). This strategy is also known as piggybacking. Piggybacking has received little formal analysis in academic literature (Dou, 2018). When platforms take advantage of opportunities to acquire consumer traffic without a partner it is often known as "growth hacking". A famous example of this is when Airbnb made a publish listing integration to Craigslist and leveraged the network to acquire more traffic to their site. Another example is Paypal's charity robot that bid on eBay auctions but demanded to pay with PayPal.

5. *Side switching* is a strategy where platforms aim for users who can fill both sides at the same time. This assumes that neither side's service requires highly specific knowledge. Etsy is a platform for buying and selling handmade goods and successfully applied this strategy. People who make handmade goods are also the ones most willing to buy them.

Strategies can be divided further into being either sequential or simultaneous.

2.4.2 Sequential

A sequential entry strategy targets one side first in onboarding. It often involves an incentive for first joiners. For example, free premium accounts for a given time or other material subsidies (Parker, 2016). These subsidies convey confidence about the success of the business and shape expectations among first joiners (Evans and Schmalensee, 2016). However, if the sequence before onboarding the second side is too long or even unsuccessful, the first side becomes impatient and will leave the platform again. One possible solution to minimize the impatience can be to openly communicate when the platform plans to onboard the second side.

2.4.3 Simultaneous

A simultaneous entry strategy targets both sides at the same time. The big-bang strategy for example is a simultaneous onboarding of producers and consumers by classic push market means. This triggers a high volume of awareness towards the platform. However, this often requires a larger marketing budget which for startups is often not feasible.

It is suggested that platforms with switching sides implement a simultaneous entry strategy while platforms with non-switching sides implement a sequential entry strategy (Schirrmacher, 2017). Below is a figure of the types of strategies discussed (as well as the seeding and big-bang strategy).

Strategy	**Description**	**Order of Entry**	**Source**
Single-marquee	Valuable users that attract others	Sequential	Evans 2009
Platform staging	Offer benefits to one set of users	Sequential	Evans 2009
Subsidy	One user group receives financial guarantees	Sequential	Parker 2016
Seeding/Self-supply	The platform itself creates value units by acting as a first producer to attract a set of potential consumers	Sequential	Evans and Schmalensee 2016
Piggyback	Addressing users on existing networks	Simultaneous	Parker 2016
Big-bang	Simultaneous onboarding of producers and consumers by marketing means	Simultaneous	Parker 2016

Table 2.1: Launch strategies and the order of entry

An action-based simulation of a two-sided fictional platform was done by Stummer (2019) in order to investigate different launch strategies. By attempting to measure the word of mouth effect (WOM) they found that that the preferable strategies seem to be self-supply for most safety, subsidization for minimum expenditures, and marquee users for maximum growth (Stummer, 2019). However, the authors warn that the results should not be generalized.

To illustrate two different strategies let's look at two prominent companies. Facebook had a carefully crafted launch strategy. They were able to maintain the quality of their network while the user base grew because they required early adopters to have a valid student (.edu) mail and universities usually issue only one mail address per student. Also, Facebook did not open to new colleges before at least twenty percent of the student body showed interest. This way they maintained sustainable growth over time. Facebook was built on top of existing social networks people had in the real world. They gave out invites that you could send to your friends so when high school students wanted to join they would have to be invited. This was in stark contrast to MySpace who wanted to grow at all costs (Moazed, 2016). Fake profiles, spam, and inappropriate content flourished on the platform and drove users away. Today MySpace has 300 million profiles but is a ghost town. The houses are there but there is no one home. Facebook focused on a single target group and also had elements of platform staging to it because the platform offered immediate value by providing lists that showed which students took which course. This exemplifies how network management is crucial for the sustainable long-term growth for an MSP.

Chapter 3

Methodology

Software engineering research is concerned with the development, operation, and maintenance of software products. This thesis examines multi-sided platform startups and the development of their platforms with a focus on the core interaction. Also, the thesis explores which current launch strategies these MSP's use and tries to explore the relationship between software development and the chosen launch strategy. This chapter deals with the methodology of the research.

The rest of the chapter is structured as follows: Section 3.1 explains the chosen research method. Section 3.2 revisits the research questions. Section 3.3 first describes the case selections and then the interview process. Section 3.4 clarifies the data collection, while Section 3.5 describes the analysis methods. Section 3.6 explains the 6W3H framework. Section 3.7 defines the PBMC.

3.1 Research Method

Several data generation methods were considered. A qualitative approach fits best when looking at startups because it is an opportunity to find unique data. Surveys were also considered as they are good for generalization of results, but gathering a large amount of multi-sided platform startups proved challenging and quantitative data is not as relevant in answering the research questions. Observations were considered too time-consuming and were not feasible due to external global events prohibiting physical meetings. Semi-structured interviews were considered a feasible option as they allow for rich data collection and the openness allows for discussion and emergence of new topics. When it comes to conducting interviews, observations, and other conversations there is a need to capture the whole context in its natural setting (Fontana A, 2000). A case study is used as a research strategy to study real-life contexts and events that are not clearly revealed. A multiple case study approach was applied in order to gather perspectives on several types of multi-sided platforms. The overall study is regarded as more robust with a multiple case study (Herriott, 1983). The study is of exploratory nature as it seeks to create knowledge

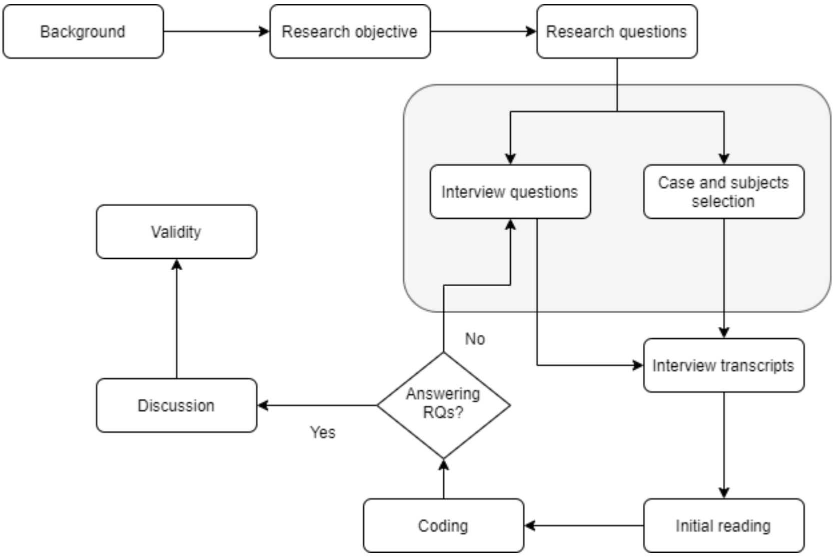


Figure 3.1: Research process

by investigating events and actions of those who experience them (Oates, 2005). Figure 3.1 illustrates the steps in the research process.

3.2 Research Questions

In this section, the main research questions are revisited. From the literature review it was revealed that little is known as to how multi-sided platforms startups develop their platforms. More specifically, how startups develop their core transaction. In addition, there is a gap in the identification of which launch strategies are used by platform startups and if there is any relationship between the software development and the launching phase. There is no doubt that the chosen technology stack affects the technological architecture of the platform and the possible performance. In this thesis, the technology stack is viewed as not relevant in answering the research questions and is therefore not explored. With the core transaction being such an important factor for the success of multi-sided platform startups, it motivated the following research questions:

1. How do multi-sided platform startups develop their platform in order to support their core transaction?
   (a) Which software methodologies are used in MSP startups?
   (b) Which requirement engineering practices are used in MSP startups?
2. Which launch strategies do MSP startups use?
3. How do MSP startups coordinate the development of the platform with launch?

3.3 Case and Subject Selections

When selecting a particular case to study in a case study, there are certain aspects that you might base your selection on (Oates, 2005). The Nordic market was consciously selected as a geographical location for the research. The high adoption rate of digital technology and high employment in the IT-sector are reasons why startups can have a good foundation for growth in this region. Also, there are government incentives, for example, Innovation Norway who funded information and communication companies with 506 million NOK in 2019 (Norway).

Startups that are either on the cusp of launching their platform or in the early years of operations were considered to be good candidates for investigation. Startups are often more willing to accept interviews than larger companies. However, tracking information about startups is more difficult.

In order to locate companies, a search was conducted on Crunchbase (a database of early-stage startups), thehub.io, and in existing networks. The Hub is a community platform for Nordic startups sponsored by Denmark's largest bank. They have helped Kahoot and Wolt grow, among many others (Hub, 2020). In addition, some interviewees identified other platform startups at the end of the interview session. This led to a couple of additional interviews. The search on Crunchbase was filtered on Scandinavian startups and the keyword ”platform startup”. The query returned 14 results. After initial research on each company, 8 were revealed to actually be two-sided markets. In the end, 2 of the companies were willing to be interviewed. Information on the rest of the companies came from differing news articles from Norwegian media and the social network LinkedIn.

The selection process for respondents was chosen based on five factors; availability, willingness, competence, and involvement. They needed to be available for interviewing and they had to be willing. They also need to have a certain knowledge of the project in order to be capable of answering some of the questions. Involvement meant that they needed to be involved in the project in one way or another in order for them to provide any insight, however that did not mean they had to be a contributing member to it. All interviewees were largely involved in the startup and had background knowledge on software development of the platform. Interviewees ranged from developers and founders to CTO's.

3.3.1 Interview Process

A consent agreement was sent to respondents informing them about the overall topic of the interview, the purpose of the interview, and that it would be audio recorded. Any recordings made were done with the approval of the respondents beforehand and they were informed that the recordings would not be published. The respondents were told in advance the topic of the interview and the initial set of questions. However, the question guide was not sent beforehand as it could not contain all possible follow-up questions. Respondents were asked for their personal opinion on digital platforms and their venture but the respondents will not be named. They were also free to decline specific questions or choose not to participate at all.

The interviews were semi-structured which allows for follow up questions that contribute clarity and even richer data (Doody, 2013). The interview guide consisted of three parts with around twenty questions. See the appendix for the interview guide. One pilot interview was conducted in order to see which questions worked and which needed to be reformulated. After the pilot interview, the questions that did not give satisfactory results were adjusted. In addition, some of the questions are harder for some respondents (depending on their background) to answer. After each interview, the questions were adjusted if necessary in order to gain richer data. Since interviewees had different positions and backgrounds, it was necessary to be flexible when it came to how the questions were formulated. Ten interviews were conducted in total. One startup had two willing interviewees and their differing roles gave new insights on the same questions. The interviews were conducted in a relatively (to the overall project length) small time window. The average interview lasted roughly around 45 minutes per respondent and was conducted over a two month window.

3.4 Data Collection

The selection of an appropriate method for data collection is essential for ensuring credibility for the content analysis (Elo, 2014). The main method of data collection in this thesis was done through interviews, identified as an efficient method for answering research questions in case studies (Oates, 2005). A challenge with qualitative interviews is that they often generate rich and broad data, but can be less precise. The interviews were conducted mainly in Norwegian, due to convenience and ease. The resulting translations were ensured to "express all aspects of the meaning in a manner that is understandable" (Larson, 1991). This implies that the interviews were not translated word for word, but that the context and meaning were preserved.

Here is an overview of the interviewed platform startups.

Case	Place of Interview	Role	Duration
Doctors and patients	Google Meet	CTO	52 min
Performers and gigs	Google Meet	Founder	35 min
Local associations and members	Google Meet	Developer	40 min
Artists and gigs	Google Meet	Developer	40 min
Events suppliers and organizers	Google Meet	Founder	45 min
Authors and freelancers	Google Meet	CEO	50 min
Local shops and customers	Phone	Founder	45 min
Products and re-sellers	Phone	Co-founder	45 min
Products and re-sellers	Google Meet	CTO	70 min
Sports clubs and members	Google Meet	CTO	50 min

Table 3.1: Interviews

3.5 Data Analysis

The data analysis was conducted with a thematic approach to qualitative content as it produces more in-depth analysis. Thematic analysis allows for deeper insight into the data and helps in analyzing it fluently as the data collection and analysis are done continuously with each other (Braun, 2006). An initial reading of each interview was done and then they were transcribed shortly after. The interviews were then analyzed by creating “open codes” which were used to generate key categories (Charmaz, 2001). As the data collection and analysis process was done simultaneously, adjustments to the interview guide were possible and it resulted in more data on the categories that were emerging from the prior interviews.

Coding is the most effective and easy way to interpret and go through qualitative data (Graneheim, 2004). The quality and reliability of the research increases because there is transparency in the step by step process of how the data was interpreted. Further, a tool called NVivo that helps with categorizing data into several nodes was used. Descriptive coding is a technique that involves labeling the data with a word or two in order to identify the basic concept. This goes a long way to categorizing the data. A total of 20 codes were generated from 218 references.

3.6 6W3H Analytical Framework

This section explains the framework that aids in contextualizing the multi-sided platform startups. There a lot of factors that play in so a framework is needed as a basis for conducting quality research. Theoretical frameworks are key to the structure and direction of the thesis, much like a blueprint to a house (Grant, 2014). As such, there has been an emphasis on finding and utilizing frameworks that fit the motivations of the thesis. Furthermore, they should fit the logical flow of the work, contribute to finding insights, and organize the data as indicated in the methodology.

This framework is built on the ”Five W’s and How” method that is widely used in for example journalism. It can help aid in the understanding of the complete story on a subject. Beyond journalism, characterizing contextual elements is commonly used in problem analysis, project management, and software engineering research. Nguyen proposed the 6W3H for building MVPs and argues that a build-measure-learn loop can be planned, visualized, and managed by such a framework (Nguyen-Duc, 2020). He defines an MVP as ”... a proxy of the final product that requires the least effort to develop but obtain maximum learning. MVP is useful for project planning, product development, fundraising, and communication” (Nguyen-Duc, 2020). In a broader context, platforms are just more refined MVP’s. Therefore the framework is also applicable to multi-sided platform startups. The 6W3H is an analytical framework that can help capture the comprehensive context of platforms. That is why it is chosen in this research. The continuous awareness and analysis of the context elements would give a useful means for visualizing and managing the evolution of a multi-sided platform startup.

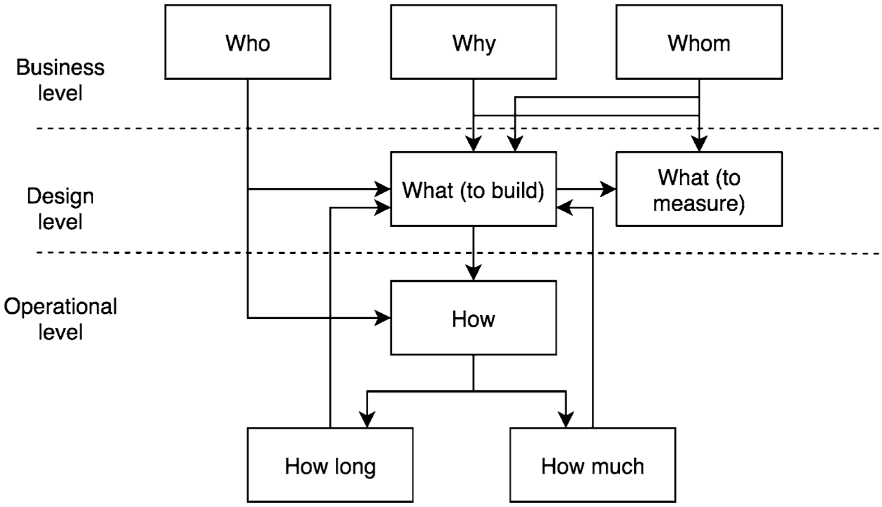


Figure 3.2: 6W3H Framework

3.6.1 Descriptions of the 6W3H Framework

In this section, the mapping of the analytical framework to the multi-sided platform startup context is explained.

What: The framework splits this question into two - what to build and what to measure. What platform to build is based on identifying the core transaction. For multi-sided platform startups, relevant data to consider is the number of users, converted users, and which features are being used. However, in the beginning, there is little to no data, so measuring is naturally challenging. In later stages, the platform will have more data to measure and can use it to develop useful features that are customer-driven.

Why: The framework mentions the need for several short-term why questions that should be answered. In a platform context, this will pertain to why this product is needed for the producer and consumers. Startups should constantly ask themselves which features are must-have and which are nice to have. This relates to the requirements prioritization phase.

Who: The next question is who is going to build the platform. The framework explains that in startups the initial team builds the product with the given set of competencies. As the startup evolves, this often changes rapidly as new knowledge and team members are being acquired. Startups need to decide if outsourcing the development or building it in-house is the best option.

Whom: The Whom question asks who might influence the product in any way. This can be external stakeholders like investors. Multi-sided platform startups rarely have an explicit customer, so the requirements are often invented or market-driven.

When: Early platforms might miss on their core transaction as the domain is highly unknown. The when question is crucial as startups have limited time and money to test their business ideas.

How This question pertains to the process and methods for realizing the product. In this context, it is the platform's core transaction. As the focus of this thesis is on the software methodologies for supporting the development of the core transaction the how question is in the center of the framework.

This thesis focuses on the what, why, and how questions of the framework because they provide the best context for answering the research questions.

3.7 Platform Business Canvas Model

The definition of a platform in this thesis relates it to being a business model. Therefore, it is appropriate to mention the platform business canvas model. It is rooted in one of the most widespread business model frameworks over the last years - "The Business Model Canvas (BMC)". The BMC is a theoretical framework, that according to its creators, aims to help answer the question of how we can, "turn visionary ideas into game-changing business models that challenge the establishment" (Osterwalder, 2010).

The Platform Business Model Canvas (PBMC) is a modification of the Business Model Canvas. It was chosen based on its ability to identify the core transaction for platform businesses. This will help in identifying the core transaction of the platform startups. In addition, it should be noted that Lean Startup is heavily influenced by the BMC and that is why it was chosen as an appropriate lens to view multi-sided platform startups through. Startups have a business model that will constantly change due to certain factors or strategic pivots. As mentioned in 2.3.1, this thesis focuses on the core transaction part of the PBMC. This means figuring out who the consumers and producers are, what the value proposition is, and how it is captured. A platform must support the core transaction in a good way in order to be successful. Below is a figure of the PBMC from Korrhonen 2017, with the core transaction elements in focus.

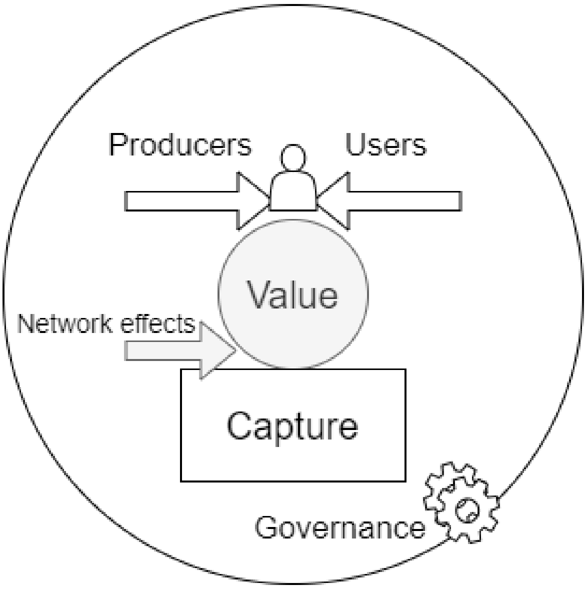


Figure 3.3: Platform business canvas model

The metrics are an important part of the business model as well because it's what the business uses to measure the success or reception of new releases. Therefore, defining what to measure is important for multi-sided platform startups and it related to one of the two questions from the What part of the 6W3H framework. The PBMC aids in defining the core transaction of a multi-sided platform and it complements the 6W3H in that regard. The 6W3H framework provides the bigger context of the whole platform in terms of what, why, and how to build it. This thesis applies the PBMC to the selected MSP's. The 6WH3 was instrumental in framing the interview questions.

Chapter 4

Results

In this chapter the findings from the multiple case study of nine multi-sided platform startups from the Nordic region is presented.

Section 4.1 explains the case descriptions. Section 4.2 answers the first research question concerning platform development. Section 4.3 concerns platform launch. Section 4.4 describes the results for research question number three. Section 4.5 is a general discussion and elaboration of the results. Section 4.6 describes the identified validity threats for this research.

4.1 Case Descriptions

The following table provides an overview of the platform startups. The categorization of the startups are based on the definitions from 2.3. In addition, the year and location of each startup is provided.

Case	Sides	Stage	Year	Location
Startup 1	Doctors and patients	Scaling	2017	Norway
Startup 2	Artists and venues	Seeding	2015	Finland
Startup 3	Local associations and members	Growing	2010	Norway
Startup 4	Artists and venues	Seeding	2018	Norway
Startup 5	Event suppliers and hotels	Seeding	2017	Norway
Startup 6	Authors and author services	Growing	2017	Norway
Startup 7	Local shops and customers	Seeding	2015	Norway
Startup 8	Brands and mediators	Seeding	2019	Norway
Startup 9	Sport organizations and members	Growing	2015	Norway

Table 4.1: Case descriptions

Here follows short case descriptions for each platform startup and their respective core transaction, identified using the platform business model canvas. The descriptions are made using the 6W3H framework with focus on the what, how and why questions.

Startup 1
S1 connects available doctors with patients. The company was founded in 2017 and is one of the larger companies interviewed. Around launch they had 1200 patients per week. They want to ""make it easier for patients to book, and ... simplify search costs." In addition to being a facilitator of the connection through both a web site and mobile application, they have their own physical clinics. Therefore this company is not only a pure inter-mediating platform because it also provides stand-alone medical services. The idea is to provide anyone with an alternative to possible long wait times at the doctor's office. They build the platform in-house with own developers and have successfully raised a substantial amount of capital in order to continue expansion.

Users	Producers	Value proposition	Value capture
Patients	Doctors	Simplifying the process of booking doctor's appointments and providing medical service in person or over video	Pay per appointment or consultation

Table 4.2: S1 core transaction

Startup 2
S2 is in a concept phase and aims to connect all types of artists, performers, and magicians with organizers such as bars, restaurants, events, and other establishments. After initial surveys, the hypothesis were confirmed. Establishments would like an easier way to connect with types of performers that they currently have to search for and contact on their own. The startup is based in Helsinki, Finland and was founded in 2015. However, the startup is currently on hold as "we just did not have time to take the service forward". The startup showed promise, launched and had paying customers. The platform was developed in house and with focus on rapid prototyping. The capturing of value happened through transaction-based fees of every booking.

Users	Producers	Value proposition	Value capture
Organizers	Performers	Simplifying the process of booking performers for events	Per booking

Table 4.3: S2 core transaction

Startup 3
S3 connects local associations with their members. They aim to provide an easy overview of activities and messages from your association. An example is sport organizations. Through the platform coaches can communicate and connect with their team and have everything in one place. They have a web and mobile application and around 33000 active users per month on the mobile application (both iOS and Android). ”We are very much a startup because we are in a continuous growth period and are still trying to establish ourselves.” The platform has several features and coaches pay per module they wish to use. Members do not pay anything for using the platform. The platform is developed in-house with one small development team.

Users	Producers	Value proposition	Value capture
Members	Coaches	Simplifying the process of local organizations communication with members, mostly sport organizations	Pay per module used

Table 4.4: S3 core transaction

Startup 4
S4 focuses on connecting artists and organizers, thereby simplifying the whole booking process. Instead of focusing on performers in general like S2, they focus on young up and coming Norwegian artists. They started as a newsletter featuring new, exciting and young talent around 2018. Now they are on the cusp of launching their booking platform where artists sign up and can be booked by mainly professionals or larger festivals. The payment structure is undecided, but with plans on either having a sort of credit system or pay per booking. The platform is developed in-house, mainly by students and there have been several different programmers on the project. As of now there are two main developers on the project working on it part-time.

Users	Producers	Value proposition	Value capture
Organizers	Artists	Simplifying the process of booking artists for events	Undecided

Table 4.5: S4 core transaction

Startup 5
S5 connects event suppliers like caterers, artists, and venues with willing organizers, mainly businesses. It was founded in 2017 born on the idea that hotels (and businesses) do not have an easy time booking different suppliers for events. It is difficult finding individual suppliers, and at the same time suppliers want easier access to customers. They have listed 55 possible event suppliers on their website and the platform is set to launch in the coming months. There are two main developers working on this in-house and the platform will be web-based. They plan on having both a transaction based fees and subscriptions.

Users	Producers	Value proposition	Value capture
Organizers	Event suppliers	Simplifying the process of organizing business events by gathering event suppliers in one place	Pay per transaction and subscriptions

Table 4.6: S5 core transaction

Startup 6

The interviewee of S6 describes the idea behind their platform with the following, “Something was missing in the market and we want to empower authors to give out their own books that can compete with the established publishers.” S6 connects authors with people who offer author services like editing, design, and book marketing. The people who offer these author service are often freelancers. By listing various services on their platform, it simplifies the process authors have to go through in order to publish their book. The current process is much dictated by the larger publishers terms and conditions. The company was founded in 2017 has just short of 1000 registered users. The platform is developed mainly in-house, but with plans on also utilizing external developers. They platform is web-based and the company is currently working on the growth plan of their business.

Users	Producers	Value proposition	Value capture
Authors	Freelancers	Simplifying the process of publishing books by gathering author services in one place	Pay per transaction

Table 4.7: S6 core transaction

Startup 7

S7 aims to connect local shops with a larger customer base. Conceptually, it is a platform that will be a contrast to the big warehouse distributors such as Amazon. The startup wants to target local shops and farmers that want to sell online but struggle with delivery and their online presence. By offering a way for these shops to list their products in one place and bringing potential customers to the table, the startup hopes to achieve this. The company focuses on the shops own branding and identity and also aims to provide delivery services. The company was founded in 2015 and are currently in a test phase, while simultaneously trying to gain commercializing support from the public sector. The platform is developed in-house and is web-based.

Users	Producers	Value proposition	Value capture
Customers	Local shops	Connecting smaller shops with a larger customer base by providing them with a storefront to sell online	Undecided

Table 4.8: S7 core transaction

Startup 8

S8 aims to connect product suppliers with re-sellers through various sales channels. According to the interviewee the core transaction is ”...that the supplier and re-seller can connect to each other, making it easy to post these products on online channels.” The idea behind order fulfillment is built on the concept of drop-shipping, where the re-seller never actually handles the product. The company was founded in 2019 and they launched their platform in May 2020. The platform is web-based and is developed with a contracting partner.

Users	Producers	Value proposition	Value capture
Re-sellers	Product Suppliers	Enabling connection between product suppliers and re-sellers with a complete platform for managing different sales channels	Pay per transaction

Table 4.9: S8 core transaction

Startup 9

S9 aims to connect local sport clubs with their members. They wish to simplify the everyday life of people who manage sports organizations (or clubs) by helping them to build the club, culture and team. They are currently in a growing phase and the company was founded in 2016. The platform was initially developed with a student consulting team, but the company now has its own in-house development team. It is free to use for members of the clubs and the platform is a mobile application.

Users	Producers	Value proposition	Value capture
Members of sport organizations	Sport Clubs	Simplifying the communication process and management of sport clubs	Subsidy

Table 4.10: S9 core transaction

4.2 Platform Development

This section aims to answer the first research question.

> *How do multi-sided platform startups develop their platform in order to support their core transaction?*

This will be done by answering the two sub-questions followed by an overall discussion of the relevant results. For all the research questions the results will be presented and discussed at the same time. However, as research also uncovered challenges and topics deemed relevant to mention, a general discussion is presented in section 4.5.

4.2.1 Which software methodologies are used in MSP startups?

Agile methodologies like Scrum, Kanban and principles from Lean are used in various degrees by MSP startups in this study. The results are in line with previous research on software startups, which is that startups do not follow any rigid methodology. Not surprisingly, it is similar for multi-sided platform startups. More mature startups have a firmer grasp of agile methodologies while younger startups are in a more exploratory phase.

> S2 - *"Working with a client, it is a must to use (software methodologies). With limited time and resources it was too slow to do this. The daily stand-ups and backlog - it can be counted as waste in many situations."*

> S6 - *"We haven't used any concrete methodology. It's been more...we have a long term plan on it. But now it's more ad hoc, and trying to solve the more pressing issues at hand in any given time."*

Given the limited amount of funding and time, startups need to be fast. MSP's value speed and generally do not use any concrete methodology. However, the importance of having a working structure is recognized.

> S6 - *"It's not always easy to understand each other side. What is stopping us from getting 100 new customer overnight? We brainstorm and pick the three to five most important tasks and work in weekly sprints. We are working more and more structured with it. In the beginning it was more ad-hoc and not systematic. In hindsight, not everything was the best solution."*

> S8 - *"Methodologies can help us, but for a platform - the most important thing is that the system is modular and good planning helps us along the way. Without any processes I think it is impossible. A platform always needs new functionalities and modularity, therefore it is extra important."*

Even though some startups view certain aspects of agile methodologies as wasteful, it can also save money.

> S8 - *"We were lucky to have a good team, but it saves money to have some practices when we are 4 engineers working."*

Scrum

Scrum is based on short development iterations and uses sprints, daily stand-ups, sprint reviews and a designated scrum master to lead the charge. Most of the startups had heard of Scrum and acknowledged that it could be helpful, but several also viewed it as too formal.

> S2 - *"We are part of the Scrum thinking but it can be very formal. It is better to just do. We just did not have the luxury of doing the formal protocols. We did not do sprints. We just started building it up."*
>
> S3 - *"It's such a small team that we do not do daily stand-ups. It's a waste of time... (we use) a very specific type of scrum, we cherry-pick the practices."*
>
> S7 -*"(We) touch on the scrum methodology and use agile work methods."*

Startups in the seeding phase were more relaxed when it came to scrum and sprints. Some did not have deadlines at all, and the founder acted as the main stakeholder.

> S4 - *"We don't have any deadlines, but as long as there is some progress the founder is happy."*

Most of the startups used sprints in some form, generally with 2 week intervals.

> S3 - *"We do sprints in 2 week intervals."*
>
> S5 - *"We have had sprints in short periods and more intense short periods of work"*
>
> S6 - *"We have meetings and a scaling workshop with board and our team."*

Other startups had more flexible meeting hours, but found it helpful to have specific goals which Scrum does specify.

> S6 - *"It really motivated the team to have specific goals to work towards. We haven't been good at setting deadlines. For example having a seminar where we had pressure to deliver new features really motivated us."*
>
> S7 - *"We have had milestones in terms of deadlines and important chats and stand-ups."*

Kanban

Kanban is a principle for planning practices and there's a wide array of tools available that can implement it. These include Trello, Azure, and GitHub Issues. Keeping track of who works on what and which features are currently under development is important and has showed itself to be helpful. Some startups use these tools and found this practice as the most important one, while others were not as aware of the principles of Kanban.

> S3 - *"The most important practice is really having an internal tool where we can keep track of issues, features and who does what. We use Kanban boards for this."*

> S4 - *"A good practice is that we use GitHub Issues in order to keep track of who does what and what is being worked on."*

> S6 - *"We have all tasks in a Trello board with to do items, bugs, features all in one place."*

> S7 - *"Primarily we have used Kanban to keep track of the backlog."*

Mostly all of the startups use a form of Kanban work practice. The most helpful feature of Kanban is really the overview it provides for a development team. The cost is low to create online Kanban boards and online tools significantly reduce the need for physical boards. As every developer can see who is doing what, it is easy to navigate to other tasks that are in need of dire attention. The board does not only keep track of the features but startups also used them for bug tracking and testing. Startups in the growing face found Kanban to be more problematic and instead pivoted to using more of elements from Scrum.

> S9 - *"We worked with Kanban, but experienced a lot of problems with scope creep and interruptions from customers. We started with Scrum instead in order to get full clarification with the customer about scope. The scrum master also protected the developers from unwanted bother."*

Lean

Lean methodology could be said to lend several principles to agile methodology with the introduction of MVP's and quick development cycles. Several quotes that did not address either Scrum or Kanban were often classified as being related to some lean principle. There are varying degrees and understanding of MVP. Some startups viewed MVP as a quick way to test hypothesis, but others had the understanding that it needed to be a complete product. An MVP can be as simple as post-it notes (Nguyen-Duc, 2020).

> S2 - *"We started to build it up and we just used WordPress initially. It was good enough for our purposes, but this was the most cost-efficient way of doing it."*

> S4 - *"We did not have an MVP. The founder was the sole voice of the requirements. We focused a lot on features and we did not really care about design. The idea was that we should make it nice afterwards."*

Some startups were quicker in developing prototypes and testing them in iterations while others relied more on longer development cycles. Startups that are in the seeding stage have less developed MVP's and are not as concerned with the design. There is less customer validation as well. Startups that have customer journeys and focus on rapid prototyping have a better relationship with their users.

> S5 - *"We used this prototype a lot on the suppliers. It provided us with information and we changed it accordingly. What really helped was that we had a customer journey - a long document illustrating the whole process. The suppliers looked at it and provided valuable feedback. This was done before the click-based prototype."*
>
> S5 - *"We had two main iterations, but several minor changes in between. In terms of software development as well, we test as we develop."*

Speed is valued in startups. In the seeding stage of startups manual labor was seen as a starting point for fulfilling some of the first orders.

> S6 - *"We started with a lot of manual processes because we want to develop the least possible version to test the marked...focus on not developing too much at once and releasing more often."*

Feedback loops in the MSP's were varied. Most recognized their importance. However, a lack of data was seen as a hindering factor in getting the most out of customer feedback. In addition, the notion that more customer validation could always be done was echoed by several MSP's.

> S8 - *"For our purposes, for the web page, we think about quick feedback loops. The shorter we can make these cycles, the more agile we can be. (We) try to implement these into our processes as we are in an early phase."*
>
> S1 - *"Customer feedback is always something we should do more of."*

Based on the data collected the startups were categorized as belonging more to one methodology or another. Those that did not exhibit traits of any particular methodology, but relied on some lean principles were categorized accordingly. This is shown in the table underneath.

Methodology	Startup
Scrum	S1, S7, S8 , S9
Kanban	S3, S6
Lean	S2, S4, S5

Table 4.11: Overview over startup methodologies

Development wise there is no consensus on any methodology. Kanban is used by the majority for keeping an overview of tasks. This helps in planning the development, seeing who does which task and when they are completed. It also provides a backlog of history so it is easy to trace the development path. Scrum was viewed as more formal and startups in the growing phase found it more helpful than startups in the seeding phase. All the startups are considered as working from some Lean Principle, but S2, S4, and S5 did not exhibit any particular leaning towards Scrum or Kanban.

4.2.2 Which requirement engineering practices are used in MSP startups?

Having some insight into which methodologies that are used helps answer how MSP startup developers work with the platform. It answers how the MSP startup teams work. However, it is only part of the answer, as having a clear vision of requirement engineering is crucial for knowing what to build. Requirements are either invented by the startup or gathered by potential users (market-driven) when there is no clearly defined customer. The three main themes that emerged from the research were validation, prioritization of requirements and measurement. The first section explores the validation process of the studied startups, which is part of defining the core transaction in line with potential customers. The next section focuses on how the MSP's prioritized these requirements. The last section looks at what the MSP's measure in order to support their core transaction.

Validation

A product should be validated at each stage. Validation starts with validating an initial idea, then moving on to other iterations. As startups have limited time and money it is increasingly important to be able to discard ideas that will not work quickly. MSP's vary in their validation processes. The startups in a seeding phase tend to have less validation and less focus on testing their hypothesis. Startups in the growing or scaling phase had honed in on their product further with more rigorous testing of hypothesis.

> S2 - *"Before doing anything we had five bullet points, and validated the idea with 40-50 people, talking to industry associations, bars, restaurants and performers."*

> S2 - *"We collected feedback with draft versions under. We showed the value proposition, what you would like with the platform. We used everything we could to get feedback, from LinkedIn to Facebook posts."*

The founders are instrumental in early testing of the idea. MSP startups leverage existing networks in order to test their hypothesis. However, there is a lack of customer-driven requirement engineering in some startups. In some cases, the founder was the sole voice of deciding the requirements.

> S3 - *"The founder is a former coach for this kids and identified several challenges with the current process. He used his team as an experiment in order to validate certain features and test the concept. "*

> S4 - *"You have to be in the environment to get it. There's so many emails and manual labor. After an initial market analysis, there's not a lot of people offering such a service. We target the Norwegian music scene."*

> S4 - *"We haven't done any feedback or user testing of the platform. The only form of feedback is from the founder."*

Startups in the seeding phase do not perform customer-driven validation to the same extent as startups in a later stage. Startups in the seeding phase tend to rely on the founder or another team member to decide what to develop or not.

> S4 - *"We have one on the team who has insight into how the process works and he has used his network to collect feedback from what a possible system needs."*

More often than not the MSP startups reach out to one or more of the sides they envision being potential customers. The resulting discussions form some of the requirements.

> S5 - *"We are going to market to companies, but in reality anyone can use the service. This has been the wish of the suppliers because companies pay more and are more professional than the private market."*

> S5 - *"For the suppliers we had tons of conversations with them. We went through several interview guides and we tested with a click-based prototype in cooperation with an external company. Same for the organizer side, we tested the click based prototype on them as well and used surveys."*

The discussions and efforts put into this process revealed important requirements that the MSP's had not been aware of already. The revelation of new requirements cause the MSP's to pivot.

> S9 - *"We had a good approach, interviewing coaches, the administration and other key players. It unveiled a key requirement, so we made an integration to a specific accounting system. That's where we should have focused more, not the jersey visualizations."*

> S6 - *"Initially we wanted to make it easier to make electronic books in the e-pub file format. The challenge was that no one had really asked for it because authors write in Word."*

This made S6 pivot to what is now their core transaction - offering author services to authors. The MSP's also tend to look at existing platforms in order to gather requirements.

> S5 - *"We have looked at existing platforms in combination with customer interviews."*

Some of the MSP's in seeding stage used surveys in order to gain a deeper understanding of the intended market. Startups in the later stages are inclined to focus more on their current customers and use surveys for that.

> S7 - *"Based on the marked surveys we focus on supporting the smaller local businesses."*

Working with one or more of the sides often revealed that what the team had envisioned did not work in practice. The assumptions of needing a certain feature could also come from the inspiration gained from looking at successful platforms.

> S6 - *"We cooperate a lot with the editor group and we have had both physical and virtual workshops with them. I thought for a long time we needed a FAQ, but what we made was too hidden and it was too fancy. We forgot the user side of it and made something we thought was cool."*

> S8 - *"When we started, we looked at existing platforms and how they message themselves. There's some inspiration there. We have to decide what we have to use and how. One thing is the technological tools, react, frameworks and so on."*

In terms of validating their hypothesis some MSP's were targeting both sides of their potential customers, while others only one. For those who only chose one side it is not clear why the specific side was chosen.

> S8 - *"The founder had the idea, travel to a large fashion fair in Milano, and these suppliers , we have since refined the concept. We have...tested suppliers and good feedback, but we have not tested enough on the re-seller side. This is an element of concern for the next month and the way forward - to see what works."*

Prioritization

Requirement prioritization is seen as an important filter for understanding what to build. It is about "... prioritizing the deliverables to ensure the customer's most important business needs are satisfied first" (Bourque, 2014). This is sometimes done with a priority rating that is set by the founders or the software engineers. Priorities are relative and it is therefore challenging because there is a lack of real information needed for such decisions. In addition, requirements are dependent on each other. There is no emerging pattern that MSP focus specifically on their core transaction when prioritizing their requirements. Some of the startups use the developers or founders as the main input for what to build first, while others are more customer-driven. Surveys, prototyping, and customer journeys are the practices found used in these MSP startups.

> S3 - *"What we as developers think will be most useful. We discuss this internally. It is difficult because as a user it is easy to say I don't get this, but as a developer it's difficult to pinpoint what you don't get specifically."*

> S4 - *"The founder is explicitly setting the requirements. We have had a waterfall approach where the founder has a clear idea of the most important features. We implement these first, then decide subsequently what we think is useful."*

MSP startups find prioritization challenging. There is not a lot of focus on rating the priorities so insights here are limited. Startups in the growing or scaling phase are more inclined to be customer-driven and aware of what to prioritize.

> S6 - *"I think the prioritizing is the most challenging because the feedback (we get) is why don't you have this or that. One thing is what people see, and the*

> *other is reputation wise. No one see what goes in under the hood."*
>
> S8 - *"We have meetings every week in order to find out what to prioritize. We planned the design, functionalities, and plugins needed to be done, but it was delayed 2 or 3 days. Also, the web-page needed work so we did that. Third party dependency took longer than expected so this made us realize we had to re-prioritize."*
>
> S9 - *"We thought a feature we had developed was incredibly intuitive, however, these user's did not understand it at all. (Re-prioritizing) was incredibly helpful in quickly realizing where we should do things better."*

It is unclear how much the MSP startups re-prioritize their requirements. Instead of letting developers or founders set the requirements, some startups let users be the source of inspiration.

> S7 - *"User experience has been the main focus, not as much the business aspect. We look at key points and features that are important to certain users."*

MSP's are constantly battling which features are nice to have and which are must have. In addition, founders can have a profound impact on what is included in early versions of the product. This delayed the MSP from launching the product and even though interviews with users had been conducted it did not help in requirement prioritization.

> S9 - *"We had a meeting, where we discussed important features. He (founder) laid out the reasons why it's impossible to launch the product if players could not visualize their own jerseys. It's a good example of how 'nice to have' features are often mixed and confused with 'must have' features. It's my best example of how to make unnecessary features."*

Measuring

The importance of measuring and it's role in better software engineering practices and management is widely accepted (Bourque, 2014). Measuring is an important step in the build-learn-cycle as well as in agile methodologies. Typically in larger companies there are dedicated measurement teams and the process is carefully planned. The results show that MSP startups struggle initially with a lack of access to data, but are aware of the importance of using a data-driven approach. This is fueled largely by the chicken and egg dilemma. Both startups in the growing or scaling phase had concrete plans for measuring. However, platforms in the seeding phase also had a clear vision of initial metrics worth investigating. Some MSP's used larger successful platforms as inspiration for their measuring process.

> S1 - *"Etsy went from prioritizing features it thought were cool to an estimate of the value it would provide the company. They looked at for example customer lifetime value, the development cost, and maintenance costs. We*

> *implement a feedback score. We text you a feedback schema, and the most important question here is how likely are you to recommend this service to someone else? Then we calculate the NPS (network performance score) score based on a numbering system: 1-6 will affect the score negatively, 7-8 will not affect the score, while 9-10 will affect the score positively"*

S1 rates their network and uses it to improve their service. Startups in the scaling phase showed much higher levels of customer involvement in the measuring phase.

> S1 - *"This is our main measurement of customer satisfaction, quality of the platform and measuring the customer journey. We measure this on individual doctors and all of our clinics. It ties into lean startup and the word of mouth effect, and NPS gives you a direct measurement of this effect."*

Few MSP startups have detailed plans for how they would measure their platform and use it to improve. Startups in the growing phase tended to measure the usage of their features in a larger extent than other startups.

> S3 - *"We collect a lot of anonymous data, how long a feature is used, which features etc. This is crucial for us in knowing what works and what does not."*

Several MSP's benefit from existing tools that help with data collection such as Google Analytics. However, certain factors play into the measuring process. What is measured can be influenced by the stakeholders. The MSP startups in the seeding phase are especially undecided on which metrics to use.

> S6 - *"We have an admin dashboard Google Analytics that can help us but we have not used it. We have only looked at our revenue because that is what the investors are interested in. (Going forward) we want to look at the number of new users, conversion rate of users to paying customers, and average amount per transaction."*

> S7 - *"We collect primarily through Google Analytics, we do surveys and ask what people think about the different features. Our system can help aid smaller shops with product delivery. When we get more data we can compare stores and give personalized recommendations or nudges to businesses. We will get an overview if certain shops sell the same things and if they are located too close to each other. The decision-making process will improve."*

> S8 - *"We are aware of the challenges, and have implemented Google Analytics, but we are not sure of how granular the data here is. It is tricky to measure the users and so we have to think about the parameters for each test for this strategy."*

Testing and Technical Debt

Testing is an integral part of software engineering, although not a focus area in this thesis. Testing used to be viewed as something that was done after the development phase, but has in recent years become ”pervasive throughout the entire development and maintenance life cycle” (Bourque, 2014). For startups testing is understandably not the number one priority. However, testing is an important preventive method that can also aid in preventing technical debt. MSP startups show the same attitude towards testing as other software startups. Several startups already experienced some form of technical debt.

> S3 - *“We need to look at the architecture. We wanted to implement something but because of the architecture we had to discard the feature. The architecture is good but there is room for improvement.”*

> S3 - *“We wanted to implement a file system in the app. We spent a lot of time developing and designing this feature. Alpha and Beta tests passed and it seemed great. However, when we tested on the external users, they did not get it all. We had to redesign the whole feature, back to the drawing board.”*

MSP startups applying agile methodologies continue to make the mistake of implementing features that show themselves to be difficult for adoption by the end user. In this case, the architecture also imposed restrictions on what was doable by the software team. The first creator of the platform was the sole decision-maker when it came to the architectural structure of the platform. More testing of the architecture could perhaps have prevented this restriction. Technical debt is a concept that software engineers and startup members are aware of (Apa, 2020). Just like the ad-hoc use of software methodologies, there is ad-hoc usage of preventive measure and testing in MSP startups. MSP startups show varying degrees of coding practices, but see it as beneficial.

> S8 - *“We implemented easier to read code, and after that we avoided a lot of problems.”*

This can help aid in the riddance of technical debt. ’Easier’ to read code refers most likely to the formatting and using coding standards can help in this. The startup in question used the ESLIN coding standard. It varied which startups focused on testing. Both seeding and growing type of startups emphasized testing. Testing referred by the MSP startups were largely concerned with customer testing, or validation of the product more so than the actual testing of code to prevent bugs and software failures.

> S1 - *“Design Thinking workshop was really helpful for us. It’s the best experience I have had of fast prototyping and validation. We had creative workshops where we tested five users every Friday. We monitored users’ reactions with a video feed.”*

> S3 - *“We have defined different target groups. We have on Alpha test group which is internal for developers. Then we have our Beta group which is the rest of the company. If approved, we have a group of 50 test users who have signed for it. If they also approve, the release is pushed out to all our users. This is done because each group provide a different perspective. Our test users have a completely different perspective than out Beta test group for example.”*

It is still unknown as to how much the MSP startups quality test their platform. However, startups in the seeding phase were aware of the importance, but due to high costs it had not been done from the beginning.

> S8 - *”When we do testing, we do a 1 week sprint. We delegate those tasks internally to the team. In an ideal world we stress test it, but in a startup setting we cannot do it. A week ago we set up a quality assurance server, and we did not have it before because it costs money.”*

Testing can reveal and prevent both fault and failures in software systems. Most of the studied MSP’s are more interested in testing their business hypothesis than spending time writing software tests.

> S2 - *”We just did not have the luxury of doing the formal protocols.”*

4.3 Platform Launch

This section answers the second research question.

> *Which launch strategies do MSP startups use?*

Most of the research on launch strategies for MSP's stem from academic literature in economics, management and information systems (Stummer, 2018). Results show that the studied MSP's use different launch strategies, but the majority use or are planning on using platform envelopment, also known as piggybacking. Below are some general thoughts the MSP startups had on their launch or planned launch. The subsequent sections explore the different launch strategies found in practice.

By applying any number or combination of launch strategies MSP startups hope to overcome the coordination problem. Some MSP's tried traditional marketing and advertising. However, as financial resources are limited, marketing can only be done up to a certain price point.

> S1 - *"Advertising to our existing base was big. However, it does cost a lot of money on google ads, snapchat, and marketing in general."*
>
> S8 - *"Being active and present is important. Advertisements on the networks we tap into is important. Good old sale, reaching out, cold-calling, just good old hustling."*

Some of the MSP's did not find the technical aspects of building a platform to be particularly challenging. All the interviewed platforms mentioned the chicken and egg problem as either being the most challenging or one of the most prominent problems.

> S2 - *"In general, in any platform it is the chicken and egg problem. It's not the technical problems, but creating the trust in the market. Platforms are not that complex, the volumes can be thousands or millions. Again, not a technical problem. It is mainly the marketing."*
>
> S6 - *"We are in the transition from concept to growth phase. On the platform we need to be able to scale. The well-known chicken and egg problem (is the main problem), how are we going to get these freelancers that offer these services to register on a whole new platform?"*

Startups have tight budgets, and even startups in growing phases did not have any marketing team.

> S3 - *"We do not have a marketing team. We have people who are customer support and they also function as sellers. The CEO travels around and has meetings with potential customers."*

Although none of the MSP's seem to be fully aware of the terms for different launch strategies, several of them applied or planned on applying strategies that are well known in literature. Whether the platform was in a seeding or growing phase mattered little. Most of the MSP's also had general strategies that did not fall into any of particular categories.

> S6 - *"We are working on a growth strategy and we have mentioned it already so many times that we need to be data-driven.*

> S7 - *"Launching a soft beta version with several products, and we see the need during these times for basic groceries."*

> S8 - *"We have been trying to learn a lot about launch strategies."*

Several MSP's mention the importance of measurement in order to tune their strategies further. They had initial ideas of what to measure, but were more unsure of what could be the most important data besides the actual number of new users. MSP startups differentiate between active and passive users. So inherently, they are aware that some users are more valuable. However, there is little focus on marquee users from the single-target group strategy. Some of the investigated MSP's showed more focus on fostering loyal users.

4.3.1 Piggybacking

This was the most prominent strategy identified being used by the MSP startups. Leveraging existing networks proved important in spreading the word for the platform. Some MSP's had partnerships while others simply used existing social networks for marketing. However, it must be noted that advertising on existing social networking does not necessarily fall within the piggybacking strategy. The platform envelopment strategy is defined as having an explicit partner and tapping into their network, with their consent. This was done by some of the MSP's and those who were in the growing or scaling phase showed the tendency to have applied this strategy successfully. In theory, if a startup taps into an existing network without the other part knowing, it is most often defined as 'growth hacking'. No particular MSP showed to be using any 'growth hacking'.

> S1 - *"We also partnered with an airline so you could earn reward points for every booking and we also used their existing network with a mailing list of 300,000 users."*

> S2 - *"We used existing social networks to market. We spent money on Facebook, newsletters, and even Twitter. We called artists and so on."*

> S8 - *"We are launching through different platforms that have large followings. That means their users potentially can become ours. For example, Shopify now initially and later other sales channels."*

The MSP's in a seeding stage acknowledged the opportunity for partnerships as well.

> S5 - *"Event bureaus that provide a complete package can also be integrated onto the platform and find more customers."*

Piggybacking is a simultaneous strategy and so it is beneficial of addressing the chicken and egg problem because it potentially attracts both sides at the same time. It proved successful for S1 as they gained many new users. However, the partnership also brought along with it unintended consequences. The company faced scrutiny from the public that airline points could be a misleading motivational factor for booking doctor appointments.

4.3.2 Subsidy

Subsidy is perhaps the most researched strategy of them all and draws naturally mainly on research in economics (Anderson, 2014). Having a low initial price point has been shown to be effective in growing user mass (Cennamo, 2013). Platforms can then later increase the price for a given or both sides. However, the idea of a ’freemium’ business model also bring along with it the introduction of price sensitivity because users might be reluctant and skeptical of fees that are introduced at a later stage. It is known that people are less willing to pay for digital services.

> S3 - *“We have a subscription based price model. You choose which modules you want and pay accordingly.”*

> S4 - *“Organizers using the platform have to pay a subscription fee. There can also be a limit on how many artists you can book, or a sort of credit system.”*

> S5 - *“Some wanted subscription based while other wanted commission based percentages per booking. We plan on offering both.”*

Most MSP’s in the seeding stage had not set their business models fully. This is probably due to the lack of data and thereby knowledge on it’s user base. Startups in the growing and scaling phase showed more adept at knowing their pricing structures. One MSP planned on having both a subscription and commission based pricing structure. This stemmed from the gathering of wishes from the different suppliers they had interviewed.

MSP’s are concerned with not just the pricing of their revenue models, but also the trade-off by integrating with a third-party payment provider. As in-house payment solutions can be complex and expensive to implement and develop, having third-party payment solutions makes a lot of sense for a platform startup.

> S6 - *“Providers take a cut as well so we try to not use that many third parties.”*

4.3.3 Single Target Group

This launch strategy can focus on either marquee users or loyal users. None of the MSP startups focused on marquee users, but several had some focus on building loyal users. Three MSP startups were identified as mostly applying single target group launch strategy.

> S3 - *“We targeted sports organizations specifically who could use the system with their coaches.”*

> S5 - *“We develop the supplier side first where they can sign up. Then we plan on targeting the organizers. The suppliers do not get any value but at the same time it does not cost anything for them.”*

MSP startup applying this strategy seem to argue that one side has no downside in joining the platform. This strategy can also tie into platform staging. S6 also went for one side first, but they provided value right off the bat for authors. That is why they are listed as belonging to the platform staging strategy.

4.3.4 Platform Staging

MSP startups can choose a single-side strategy in the form of platform staging. This is a sequential type of strategy because one side is deliberately targeted first. However, it can be a challenging strategy because it requires the platform to provide immediate value to users. Only one MSP had this strategy, but another MSP showed interest in trying it. As with all launch strategies, the MSP's were not particularly aware of which strategy they had actually chosen.

> S6 - *"We have to show them that there is a market. We need good suppliers on the platform. We solved it by creating our own bookstore and the commission is better on any of the other Norwegian online book stores. In addition, the author profiles becomes a free marketing thing for authors."*
>
> S6 - *"As long as you have the rights to the book, you can register yourself on this site. This gave us an overview of all authors with names and pictures. This helped in attracting suppliers. All of them have found us actually. We had a marketing campaign last fall, but before that, everyone came to us. We have had a luxury problem in the sense that we could choose to allow only the best services on our platform."*
>
> S6 - *"It proved to be effective to have the authors join first."*

One MSP acknowledged the value it could have for them to have such a product, but had not yet implemented it.

> S8 - *"(Idea of) single player vs. multiplayer. Single player has some sort of value to a user without the other side. For instance how Instagram used filters to enhance photos. People just used the app for photos and then later Instagram built out its network. For us it's I mean, thinking about if we can have our own web shop and give that value to initial users, so you can then leverage that link wherever, if the integration is not there already."*

4.3.5 Big-bang

This is a simultaneous launch strategy. One MSP was identified as specifically going for both sides at the same time, but it highlighted the challenges associated with it.

> S2 - *"We went for both at the same time. We had internships from different schools that would attain emails and contact artists from different Facebook pages, and use their network."*

For this startup it proved to a big marketing cost and it was not sustainable over time.

Below is a table that classifies the startups into which strategies are mainly being used by them.

Launch Strategy	Order of Entry	Startup
Single target group	Sequential	S3, S5, S9
Platform Staging	Sequential	S6
Subsidy	Sequential	S4
Piggybacking	Simultaneous	S1, S3, S6 S7, S8
Big-bang	Simultaneous	S2

Table 4.12: Overview over chosen launch strategies

4.4 Platform Development and Launch

This section explores what the results show pertaining to research question number three.

> *How do MSP startups coordinate the development of the platform with launching?*

A theme that emerged during the research was the notion of features that would help grow users. MSP startups continuously brought up the theme of having special functionality in order to help with user growth. This is not part of any particular launch strategy. If anything, it can be viewed as falling in under the category of platform staging if the feature provides value to the users without the other side being present. This is a known strategy for overcoming the coordination problem. Several MSP startups echoed the idea that 'killer' features would either be a valuable selling point or help grow the platform. This ties into the idea of network management as mentioned in section 2.

> S1 - *"We are looking at wow factors to help with user growth. A notification for immediate feedback on tracking of your health results. A lot of people are interested in their test results. No one else has this right now."*

> S7 - *"Super checkout stores that can be pre-made and checked out for users."*

For MSP's the 'wow' features are nice to have, and the results suggests that the MSP startup members find such features to be beneficial when using it as a selling point. However, the actual impact these features would have for the given startups is unknown, especially with regard to the core transaction. For S1 the core transaction is the matchmaking of doctors and patients. A feature to track your results seems initially as a good idea. It is not known the exact process of how this feature came to be included in the requirement list. As this MSP is in the scaling phase, it has already been successful at facilitating the interaction. If not, it would not have paying customers. Matching the two-sides should be the most important aspect and so the features in question should be viewed in this light. Especially startups in the earlier phases may fall in the trap of developing something that is unnecessary. By thoroughly investigating the actual need for the feature the startups will know if it actually should implement it. In an ideal world this would be easy, but in truth startups do not have full information available.

MSP startups face the constant dilemma of balancing which features to include when it comes to release of the platform.

> S6 - *"Rating is an example. It is not a must feature to scale so it's in the prioritization discussion. However, I think it can make the product easier to sell. We haven't used any money on marketing yet. We want to know we can handle the volume."*

The actual features themselves do not seem to influence the MSP's choice of launch strategy. It is not clearly known what MSP startups mean by making the product easier to sell, but it is reasonable to assume that it implies spending less money on advertisement. By having a 'killer' feature MSP startups hope to increase the word of mouth (WOM) effect. It is worth noting that none of the investigated MSP startups use the strategy of single target group with focus on marquee users. However, unknowingly this can still be the case for some. Not knowingly targeting marquee users does not mean it is not being done. However, the chances are slim. Given that research indicates that the marquee users strategy is beneficial for increasing the WOM effect it is something MSP startups should investigate further.

A topic not explored in this thesis pertains to platform stickiness or the 'lock-in' effect. This related to how companies are able to retain users on their platform. Again, the development of these features do not seem to influence the choice of launch strategy. However, it can help with customer satisfaction which ties into a positive WOM effect. If users are positively surprised over certain features it implies that they will speak warmly about the application in question.

> S1 - *"We know that sometimes a doctor will be late for an appointment. We implemented a feature in the app where the doctor can notify the patient of a late arrival with the click of a button. These measures help with customer satisfaction."*

An important part of launching which is not extensively discussed in literature is the timing of the launch. As was indicated in some of the interviews there seems to be little to no experimentation done on when the best time to launch is. This ties into development because platform owners are the ones who are in control of the process. Results indicate that MSP startups had trouble in knowing what to exactly include in a release.

> S8 - *"We are waiting to get our application approved on Shopify, (which is) a large provider of web-shops. We already had a bit of a soft launch, a non-functional platform, but no sales channels were there and no payment system."*

A launch with a platform not facilitating the core transaction is problematic as users can be turned off by the lack of value. S8 had initially a soft launch which was a launch without a fully functioning platform. Customer feedback pointed to several features that either needed to be better or were lacking. Therefore, the startup did a classic customer need pivot by investing time and resources into developing a feature that would simplify the import process of products to the platform. In addition to being able to manually create product listings, the feature would allow automatic import of products. The startup in question re-launched officially in May this year (after the research process had concluded) and so there is a lack of information on how the new feature affected the launch.

In addition, the timing of launches seem random.

> S1 - *"Right before launch we used PayEx and everything worked in the test environment. However, 2 weeks before launch we tried it and it would take 1 month for it to be approved. Should we extend our launch or just roll with it? We chose to launch and market free video consultation until the payment solution was in place and it actually worked pretty well."*

In this case it worked well. By applying a strategy of subsidy for the patient side due to unforeseen circumstances, S1 overcame the issue. Here the development was a key reason for why the subsidy strategy was chosen, or rather, the lack of having a functioning payment system in place. It applies to the other MSP startups as well, that a certain amount of randomness is present.

Measuring user feedback seems to influence to choice of launch strategy. S8 are primarily relying on a piggyback strategy, and plan on using data collected to drastically influence the strategy moving forward.

> S8 - *"We identified these strategies and picked out those that appeal to us. We are going to do two weeks of testing on each and measure the customer acquisition, type of users, how many, etc. and the idea is to double down on any that works for us."*

Allowing for frequent exploration and systematic testing of launch strategies as is implied above, seems to be a good approach in figuring out what works for the specific platform.

For MSP startups it is crucial to know and decide whether side-switching should be possible or if it is actually possible at all. In theory, side-switching is great as producers and consumers are interchangeable. However, in practice the context of the startup must be considered. For S8 a producer of a listing is in theory anyone who has something to sell. In reality most consumers do not offer a product to sell. So there are at least two distinct sides that are possible users of their platform. As Schirrmacher (2017) suggests, platform with non-switching sides should implement a sequential launch strategy. This can also include combinations of them, but more research needs to be done in order to know the effects of multiple strategies at once. The challenging part is figuring out how to measure the effectiveness, because there are multiple factors in play at once. Therefore, any applied strategy can be the source for user growth. As to the point of measuring, MSP startups seem to be open and ready to pivot at any given time. The willingness for any startup to pivot ties into the psychological aspect as well. As a startup spends more resources and time on a product it is known that it can become harder for the startup to pivot. The reluctance to pivot can doom the venture. This ties into the management of startups and the people within the organization. These ties are worth mentioning, but research in this thesis did not specifically investigate these topics.

As far as software methodologies and engineering practices go, the results do not indicate any specific correlation between them and the chosen launch strategies. Three of the MSP startups that used Scrum (S1, S7, S8) also leveraged a piggyback strategy, but this is most likely due to randomness. The way a startup works affects the speed of platform development, but not necessarily the chosen launch strategy. No other patterns between development and launch strategies were identified in the collected data.

4.5 Discussion

From the discussion with companies several interesting challenges and thoughts were revealed that are deemed relevant to mention in a more general discussion. These topics are discussed in this section.

Most of the MSP's choose piggybacking as the prominent strategy, albeit several also had elements of other strategies intertwined. Research suggests that marquee-users is the best strategy for maximum growth. Therefore it interesting that none of the MSP's showed any particular interest in pursuing marquee users. Marquee users are believed to have high impact on the usage and adoption of platforms and are not necessarily celebrities. In relation to the studied startups there can be several factors as to why this strategy is not pursued more. Firstly, it can be difficult to identify marquee users in a given market segment. Secondly, it is not given that identified 'high-value' users correctly spread a positive WOM effect in the network. Thirdly, there may not exist any particular marquee user in the market. However, it is a strategy that should be explored more by MSP startups since the results can be beneficial. The role of this study is not to act as a consultant to any of the interviewed startups, but when looking at the startups and their markets it is not unlikely that there exists several possible marquee users. For example, S3 targets people in sport organizations. These people have a network of coaches that are possible users of the platform. However, this is somewhat of a gray area as well because it depends on if one coach is more valuable than the next. The startup needs to figure out if there are perhaps marquee coaches that can influence other coaches adoption of the platform.

Some focused on loyal users, or a set of people who were tasked with getting an early version of the platform. Startups in the growing phase seemed to have more control on their customer groups and in turn had implemented measures that may help build loyal users. These MSP's leveraged Facebook and other social networks in order to invite users to participate in discussion about the platform. These groups also acted as a forum for new feature ideas and an arena for feedback. Across the board there are big variations in to what degree this validation was done. In line with theory, startups are better off with implementing these type of measures. Interestingly enough, knowing about agile methodologies and ways of customer validation did not necessarily mean they were followed. The founder is no doubt a heavy influence on the requirement list early on, but it is not advised that these hypotheses go untested. MSP startups in the seeding phase seemed more inclined to ”just develop” and take it from there. The founder also acted as the sole voice of requirements in some cases and the development team did not ask any questions. This identifies as a classic startup trap where the product value is at risk of being far away from the actual user need.

In conjunction with this topic, some startups raised an important point - the notion of software developers acting as designers.

> S7 - *"We try to move some of the artistic tasks to developers while at the same time moving some of the technological tasks to artists. It builds understanding on both sides of the aisle."*

The classic tale of software engineering is the concern of functionality versus the design process. Developing a platform, or any type of software product is undoubtedly a multifaceted endeavour that entails many disciplines - notably the mix of technological and design competencies. As this MSP also stated in the interview, software developers can sometimes disregard the importance of the platform design. Not only that, but designers also lack understanding in the field of software engineering. Therefore, the ideal software engineer is one who can design and the ideal designer is one who can engineer software.

The reliance on stakeholders is also a challenge. There is limited understanding of the platform business model.

> S6 - *"Platform as a business model, very few people understand it. It is a challenge to convey it to investors. We do not produce anything it is our users through the platform. We just have to repeat, repeat, and repeat."*

Always having to convey the same message can be tiring. This suggests that the platform business model is perhaps not widespread enough yet. There is no product in the traditional sense of a linear business. The core is truly the facilitating of an interaction. Also, a major challenge is that the market may not exist. MSP's are left to fend for themselves and not only find a market fit, but creating it. This suggests that some ideas are simply just ahead of their time.

> S2 - *"Changing the behaviour. It was so hard to even order a book from the internet 20 years ago. Today it is normal, but still difficult to buy for example consulting services via a platform. I am sure in 10 years time, it will be normal. That's the biggest thing. Is the market or people ready? In this case, they were not."*

4.6 Validity of Findings

Validity is a goal, not something that can be proven or assured with the use of specific procedures (Feldt, 2010). However, it is important to assess and evaluate the validity of any research. Several validity threats were identified with this study. As outlined in Wohlin (2000), four types of validity exist in software engineering research - construct, internal, external and conclusion validity. The four subsequent paragraphs describe the threats identified in this research respective to the four types of validity.

Construct validity is concerned with the whole research design and process. The results are limited to the selection process of available MSP startups as well as the master scope of one year. Interview questions were designed by the researcher alone and therefore it includes a higher risk of asking irrelevant questions. However, the questions were reviewed by the supervisor and external co-supervisor to gather feedback on what the most relevant questions are. In order to minimize bias and maximize flexibility, the interview process was done in an iterative manner. As information appears while the research is done it allows for question to be altered as needed in order to gain richer data. This is an important step because it allows for new categories to emerge. However, it should be noted that the research also lacks comprehensiveness in the sense that more software methodologies and engineering practices could have been investigated. However, this is a delicate balance in the form of breadth versus depth. Video meeting where held when possible as it helps in creating a meaningful connection between the researcher and interviewee. COVID-19 did not allow for physical meeting in the startup's environment. However, the question is whether a physical meeting would have contributed any richer data than a video meeting.

Only having one interview per company except for S8 (where two people were interviewed) provided only one perspective from within the startup. This limits the data being collected. In addition, going back in time is always challenging for research because as humans we have a tendency to forget finer details over time. It is proven that contemporary studies can yield richer data, as the experiences are fresh in the interviewees mind. This study was contemporary in nature as it investigated current multi-sided platform startups. However, certain questions were naturally calling for the interviewees to reflect on previous experiences. In some cases, these experiences were further back in time and so the resulting data is probably incomplete.

Internal validity is about minimizing bias within the research itself. The focus was on finding people with knowledge on development processes. Interviewees ranged from CTO's, founders and developers. By focusing on people close to the development process, the goal was to get more accurate representation of reality. As decisions and conclusions concerning the research are dependent on a single researcher, there is naturally bias introduced along the way. To minimize inaccuracies the transcriptions were made shortly after each interview as the event was still fresh in the researcher's mind.

External validity is concerned with the generalizability to similar environments. This means if the research can be generalized in other environments. Even though the study focused on Nordic startups, more out of convenience than anything else, it is reasonable to assume that other researchers can find some benefits of this study. By investigating the types of MSP startups used in this study (as described in section 4.1), the results can be seen as transferable to other similar businesses.

Conclusion validity is about capturing the actual meaning of the interviews. By having most of the interviews in Norwegian means that there is a risk of losing some of the original meaning when translating to English. In addition, the transcriptions were not made word for word as informal oral language is quite different from written language. However, by having interviewees agree to audio capturing of the interviews, it ensured that the researcher could double-check transcriptions with said audio. In addition, collecting data on the full scope of a startup in a 45 minute interview is challenging. In order to capture more data the researcher consulted the startup's website where possible, social media and search engines.

Chapter 5

Conclusion

This chapter summarizes the research process and the main findings. It ends with thoughts on possible future works and experiences from the research process.

5.1 Summary

Multi-sided platform startups face numerous challenges, notably the chicken and egg dilemma, as well as limited time and resources. The objective of this master thesis was *to understand the ongoing software development practices, engineering activities, and launch strategies in MSP startups. In particular, how platform startups support and understand their core transaction.* This was done by investigating 9 multi-sided platform startups in the Nordic region through conducting semi-structured interviews. The interviews were transcribed and coded into categories. These categories were used to answer the research questions. In order to contextualize the MSPs, the 6W3H and PBMC framework were used.

This thesis provides the following main findings:

- MSP startups use **differing and various combinations** of agile methodologies and requirement engineering practices in order to develop their core transaction.
- MSPs in mature stages show more adeptness of using agile methodologies than younger startups.
- MSP startups should emphasize **requirement gathering and validation** to a larger extent, especially those in the seeding stage.
- MSP startups are **generally unaware** of their chosen launch strategies.
- MSP startups view the development of 'power features' as a way to solve the coordination problem, in addition to utilizing **piggybacking** as the main launch strategy.

- The results indicate **no particular correlation** between chosen development practices and the choice of launch strategy.

5.2 Future work

This research provides background on MSP startups and how they develop their platforms. It would be interesting to investigate other MSP startups that fall into a related category and see if the results are similar. Also, further research is required to map which launch strategies are the most beneficial for certain types of MSPs. This research found piggybacking as the preferred launch strategy, but this may not hold in other investigations. Another promising research area could be to look deeper at MSPs core transactions and how startups can get this transaction right as fast as possible.

5.3 Research process experiences

Similar to how a startup needs to pivot at critical times, so is the process of writing a master thesis. Digital platforms were the main focus area from the beginning. However, the initial idea was to work with only one MSP startup, and the research area was different. Instead of focusing on the core transaction of multi-sided platforms, the focus was solely geared towards launch strategies and overcoming the chicken and egg problem. The research area seemed promising, albeit it was more focused on marketing streams of research. As time moved on, it became clear that the launch of the platform in question would not fit the timeline of this thesis. Therefore it was necessary to pivot and formulate broader and improved research questions that would be of value.

The idea was to focus on the unique context of software startups and use relevant theory and apply it to multi-sided platforms. Since MSPs play such an important role in our society and will continue to do so, it was warranted as an interesting avenue to explore further. Little existing research was found on multi-sided platforms in the Nordic region. This new area of investigation required the inclusion of several platform startups, making a multiple case-study appropriate. The switch to include several MSP startups can be seen as largely beneficial. It allowed for the results to be more applicable to MSP's as a whole. Limiting the study to one startup would surely make the results much less generalizing. It should be noted that a large amount of time was spent collecting data on the background for this thesis. The research spans several disciplines and so a whole understanding is necessary. This thesis focuses more on breadth than depth but is hopefully of interest to software startup researchers, practitioners, and other interested parties.

Bibliography

Adlin, T. Pruitt, J., 2010. The Persona Lifecycle. Morgan Kaufmann Publishers.

Ailisto, H., 2016. Is finland being left behind from the train of the platform economy? Publication Series of Government's Analysis Assessment and Research Activities .

Alves, C. Pereira, S.C.J., 2006. A study in market-driven requirements engineering .

Anderson, D., 2010. Kanban-Successful Evolutionary Change for Your Technology Business. David J. Anderson Associates Inc.

Anderson, E. Parker, G.T.B., 2014. Platform performance investment in the presence of network externalities. Inf Syst Res , 152–172.

Aoyama, M., 1998. New age of software development: How component-based software engineering changes the way of software development? International Workshop on CBSE .

Apa, C. Jeronimo, H.N.L.V.D.T.G., 2020. The Perception and Management of Technical Debt in Software Startups. Springer Cham.

Bajwa, S. Wang, X.N.D.A., 2016. How do software startups pivot? empirical results from a multiple case study. International Conference of Software Business , 169–176.

Belleflamme, P. Peitz, M., 2016. Platforms and network effects. Handbook of Game Theory and Industrial Organization .

Berg, V. Birkeland, J.N.D.A.P.I.J.L., 2018. Software startup engineering: A systematic mapping study. Journal of Systems and Software Volume 144 , 255–274.

Binken, J. Stremersch, S., 2009. The effect of superstar software on hardware sales in system markets. Journal of Marketing , 88–104.

Blank, S. Dorf, B., 2012. The startup owner's manual. Academy of Management Journal , 821–843.

Bosch, J. Ljunblad, J.O.H., 2013. The early stage software startup development model: A framework for operationalizing lean principles in software startups. LESS 167 , 1–15.

Bourque, P. Fairley, R., 2014. Guide to the software engineering body of knowledge (swebok version 3.0). EEE Computer Society Press .

Braun, V. Clarke, V., 2006. Using thematic analysis in psychology. Qualitative research in psychology 3 , 77–101.

Carmel, E., 1994. Time-to-completion in software package startups. Proceedings 27th Hawaii International Conference on System Sciences (HICSS) , 498–507.

Cennamo, C. Santalo, J., 2013. Platform competition: strategic trade-offs in platform markets. Strategic Management Journal , 1331–1350.

Charmaz, K. Mitchell, R., 2001. Grounded theory in ethnography. Handbook of ethnography, Sage publications , 160–174.

Choudary, S., 2015. Platform scale: How an emerging business model helps startups build large empires with minimum investment. Platforms Thinking Labs .

Crowne, M., 2002. Why software product startups fail and what to do about it. Proceedings of International Engineering Management Conference (IEMC) , 338–343.

Dahlstedt, A., 2003. Study of current practices in market-driven requirements engineering. Third Conference for the Promotion of Research in IT at New Universities and University Colleges in Sweden .

Doody, O., .N.M., 2013. Preparing and conducting interviews to collect data. Nurse Researcher , 28–32.

Dou, Y. Wu, D., 2018. Platform competition under network effects: piggybacking and optimal subsidization. SSRN , 1–39.

Eisenmann, G. Parker, M.V.A., 2006. Strategies for two-sided markets. Harv. Bus. Rev. , 92–101.

Eisenmann, G. Parker, M.V.A., 2011. Platform envelopment. Strateg. Manag. J. 32 , 1270–1285.

Elo, S. Kääriäinen, M.K.O.P.T.U.K.K.H., 2014. Qualitative content analysis: a focus on trustworthiness. SAGE Open .

Evans, D.S., Schmalensee, R., 2016. The Matchmakers: The New Economics of Multisided Platforms. Harvard Business Review Press.

Evans, D. Schmalensee, R., 2010. Failure to launch: Critical mass in platform businesses. Rev. Netw. Econ .

Feldt, R. Magazinius, A., 2010. Validity threats in empirical software engineering research - an initial survey. Proceedings of the Software Engineering and Knowledge Engineering Conference .

Fontana A, Frey JH. Denzin Norman K, L.Y.S., 2000. Methods for collecting and analysing empirical materials. Handbook of Qualitative Research , 632–644.

Ghazawneh, A. Henfridsson, O., 2015. A paradigmatic analysis of digital application marketplaces. Journal of Information Technology , 198–208.

Giardino, C. Abrahamsson, P.W.X., 2014. Why early-stage software startups fail: A behavioral framework. Lecture Notes in Business Information Processing, vol 182 , 27–41.

Graneheim, U. Lundman, B., 2004. Qualitative content analysis in nursing research: concepts, procedures and measures to achieve trustworthiness. Nurse Education Today Volume 24 , 105–112.

Grant, C. Osanloo, A., 2014. Understanding, selecting, and integrating a theoretical framework in dissertation research: Creating the blue print for your house .

Hagiu, A., 2009. Multi-sided platforms: From microfoundations to design and expansion strategies. Harvard Business School Strategy Unit Working Paper No. 09-115. http://dx.doi.org/10.2139/ssrn.955584 .

Herriott, R. Firestone, W., 1983. Multisite qualitative policy research: Optimizing description and generalizability. Educational Researcher , 14–19.

Hub, T., 2020. URL: `https://thehub.io/`.

Kenney, M. Zysman, J., 2016. The rise of the platform economy. Issues in Science and Technology , 61–69.

Klotins, E.e.a., 2019. A progression model of software engineering goals, challenges, and practices in start-ups. IEEE Transactions on Software Engineering .

Klotins E., Unterkalmsteiner M., G.T., 2015. Software Engineering Knowledge Areas in Startup Companies: A Mapping Study. Springer, Cham.

Korhonen, H. Still, K.S.M.K.M.S.A.V.K., 2017. The core interaction of platforms: How startups connect users and producers. Technology Innovation Management Review , 17–27.

Larson, M., 1991. Translation: theory and practice, tension and interdependence. John Benjamins Publishing.

Lei, H. Ganjeizadeh, F.J.P.O.P., 2017. A statistical analysis of the effects of scrum and kanban on software development projects. Robotics and Computer-Integrated Manufacturing VOl. 43 , 59–67.

Livermore, J., 2008. Factors that significantly impact the implementation of an agile software development methodology. Journal of Software Vol. 3. , 31–36.

Martin, K. Hoffman, B., 2007. An open source approach to developing software in a small organization. Software, IEEE, vol. 24 , 46–53.

Moazed, A. Johnson, N., 2016. Modern Monopolies: What It Takes to Dominate the 21st Century Economy. St. Martin's Press.

Morthen, E., 2019. Software requirements engineering: A case study of a startup .

Nazar, J., 2013. 14 famous business pivots .

Nazir, A. Raza, S.G.D.C.C.K.B., 2009. Network level footprints of facebook applications. Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference , 63–75.

Nguyen-Duc, A. Munch, J.P.R.W.X.o.P., 2020. An Analytical Framework for Planning Minimum Viable Products. Fundamentals of Software Startups: Essential Engineering and Business Aspects. Springer International Publishing.

Nguyen-Duc, A. Seppanen, P., Abrahamsson, P., 2015. Hunter-gatherer cycle: A conceptual model of the evolution of software startups. Proceedings of the 2015 International Conference on Software and System Process (New York, NY, USA) , 199–203.

Norway, I., . Innovation norway funding. URL: `https://www.innovasjonnorge.no/no/om/hvem-har-fatt-finansiering-fra-oss2/`.

Oates, B., 2005. Researching information systems and computing. Sage , 14–19.

Osterwalder, A. Pigneur, Y., 2010. Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. John Wiley Sons.

Osterwalder, A. Pigneur, Y.B.G., 2014. Value Proposition Design: How to Create Products and Services Customers Want. John Wiley and Sons.

Pandey, D. Suman, U.K.R., 2010. An effective requirement engineering process model for software development and requirements management. 2010 International Conference on Advances in Recent Technologies in Communication and Computing , 287–291.

Parker, e.a., 2016. The digital platform: a research agenda. Journal of Information Technology 33, 124–135.

Rafiq, U. Bajwa, S.X.W.L., 2017. Requirements elicitation techniques applied in software startups. IEEE Computer Society , 141–144.

Ries, E., 2011. The Lean Startup. Penguin Books Ltd.

Rochet, J. Tirole, J., 2006a. Two-sided markets: where we stand. Rand Journal of Economics , 645–666.

Rochet, J. Tirole, J., 2006b. Two-sided markets: where we stand. Rand J.Econ 37 , 645–666.

Rochet, J. Tirole, P., 2003. Platform competition in two-sided markets. Journal of the European Economic Association, Volume 1, Issue 4 , 990–1029.

Royce, W., 1970. Managing the development of large software systems: concepts and techniques. Proceedings of IEEE WESTCON .

Salminen, J., 2014. Startup Dilemmas - Strategis Problems of Early-Stage Platforms on the Internet. Turku School of Economics.

Salo, P. Ronkainen, O.W.J., 2017. Agile software development methods: Review and analysis .

Schiff, A., 2003. Open and closed systems of two-sided networks. Information Economics and Policy , 425–442.

Schirrmacher, Nina-Birte. Ondrus, J.K.T., 2017. Launch strategies of digital platforms: Platforms with switching and non-switching users. In Proceedings of the 25th European Conference on Information Systems (ECIS) , 167–173.

Stummer, C. Kundisch, D.D.R., 2018. Platform launch strategies. Business Information Systems Engineering Vol. 60 , 167–173.

Stummer, C. Haurand, M., 2019. Evaluating market entry strategies for two-sided digital platforms under competition: A simulation approach. ICIS 2019 Proceedings .

Sutton, J., 2000. Role of process in a software start-up. IEEE Software 17 , 33–39.

Tanabian, M., 2005. Building high-performance team through effective job design for an early stage software start-up. Proceedings International Engineering Management Conference (IEMC) , 789–792.

Tiwana, A. Konsynski, B.B.A., 2010. Research commentary—platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. Information Systems Research .

Tripathia, N. Klotins, E.P., 2018. An anatomy of requirements engineering in software startups using multi-vocal literature and case survey. Journal of Systems and Software , 130–151.

Unterkalmsteiner, M. Giardino, C.G.T.o.P., 2014. What do we know about software development in startups? IEEE Software .

Unterkalmsteiner, M. Abrahamsson, P.W.X.F.N.D., 2016. Software startups - a research agenda. E-Informatica Software Engineering Journal , 89–123.

Van Alstyne, M. Parker, G.C.S., 2016. Platforms, and the new rules of strategy. Harvard Business Review , 54–60.

Wertz, B. Kingyens, A., 2015. A guide to marketplaces. Vancouver: Version One Ventures .

Wilson, C. Boe, B.S.P.A.Z.Y., 2009. User interactions in social networks and their implications. Proc. of the 4th ACM European conference on Computer systems .

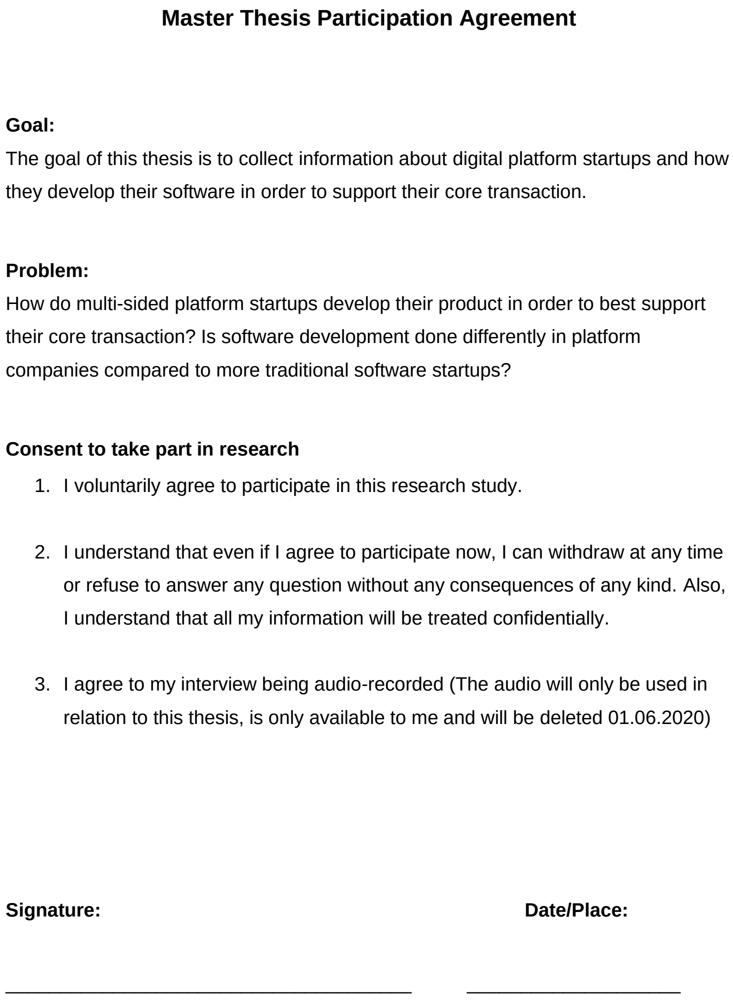
Wohlin, C. Host, M.R.P.O.M.W.A., 2000. Experimentation in software engineering: an introduction. Kluwer Academic Publishers.

Womack, JP. D, J.D.R., 1990. Machine that changed the world. Simon and Schuster.

Yau, A. C, M., 2013. Is a rigorous agile methodology the best development strategy for small scale tech startups? .

Zhu, F. Iansiti, M., 2012. Entry into platform-based markets. Strategy Management Journal 33 , 88–106.

Appendix



Master Thesis Participation Agreement

Goal:
The goal of this thesis is to collect information about digital platform startups and how they develop their software in order to support their core transaction.

Problem:
How do multi-sided platform startups develop their product in order to best support their core transaction? Is software development done differently in platform companies compared to more traditional software startups?

Consent to take part in research

1. I voluntarily agree to participate in this research study.

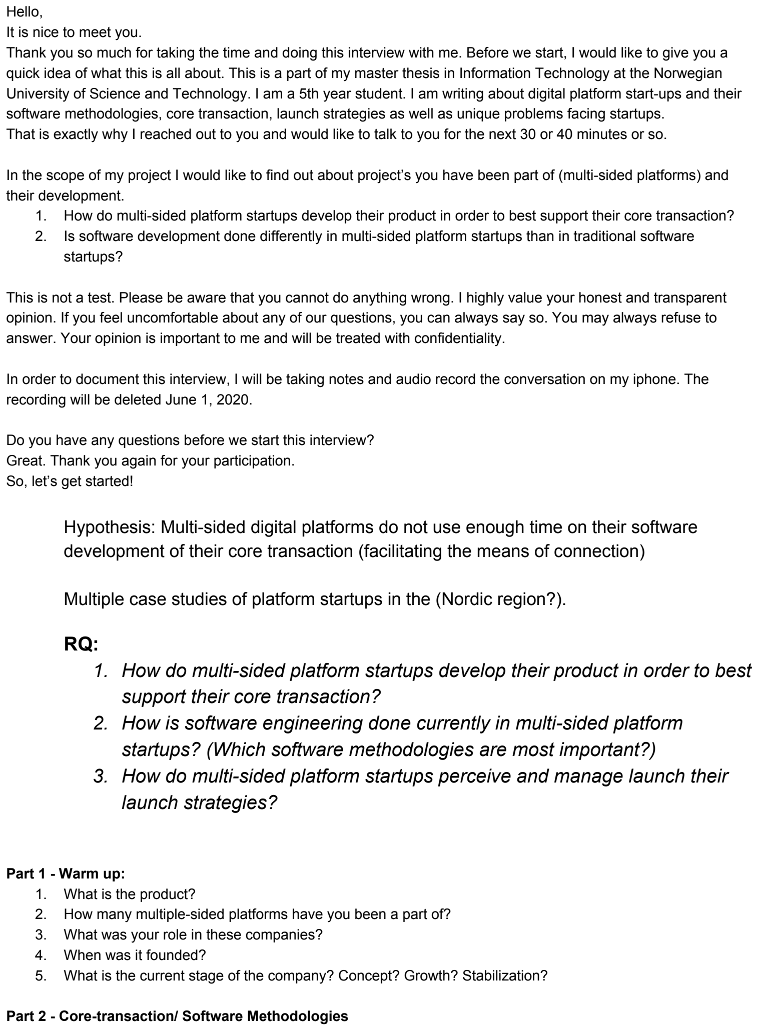
2. I understand that even if I agree to participate now, I can withdraw at any time or refuse to answer any question without any consequences of any kind. Also, I understand that all my information will be treated confidentially.

3. I agree to my interview being audio-recorded (The audio will only be used in relation to this thesis, is only available to me and will be deleted 01.06.2020)

Signature: **Date/Place:**

_____________________________________ ____________________

Figure 5.1: Consent form



Hello,
It is nice to meet you.
Thank you so much for taking the time and doing this interview with me. Before we start, I would like to give you a quick idea of what this is all about. This is a part of my master thesis in Information Technology at the Norwegian University of Science and Technology. I am a 5th year student. I am writing about digital platform start-ups and their software methodologies, core transaction, launch strategies as well as unique problems facing startups.
That is exactly why I reached out to you and would like to talk to you for the next 30 or 40 minutes or so.

In the scope of my project I would like to find out about project's you have been part of (multi-sided platforms) and their development.

1. How do multi-sided platform startups develop their product in order to best support their core transaction?
2. Is software development done differently in multi-sided platform startups than in traditional software startups?

This is not a test. Please be aware that you cannot do anything wrong. I highly value your honest and transparent opinion. If you feel uncomfortable about any of our questions, you can always say so. You may always refuse to answer. Your opinion is important to me and will be treated with confidentiality.

In order to document this interview, I will be taking notes and audio record the conversation on my iphone. The recording will be deleted June 1, 2020.

Do you have any questions before we start this interview?
Great. Thank you again for your participation.
So, let's get started!

Hypothesis: Multi-sided digital platforms do not use enough time on their software development of their core transaction (facilitating the means of connection)

Multiple case studies of platform startups in the (Nordic region?).

RQ:

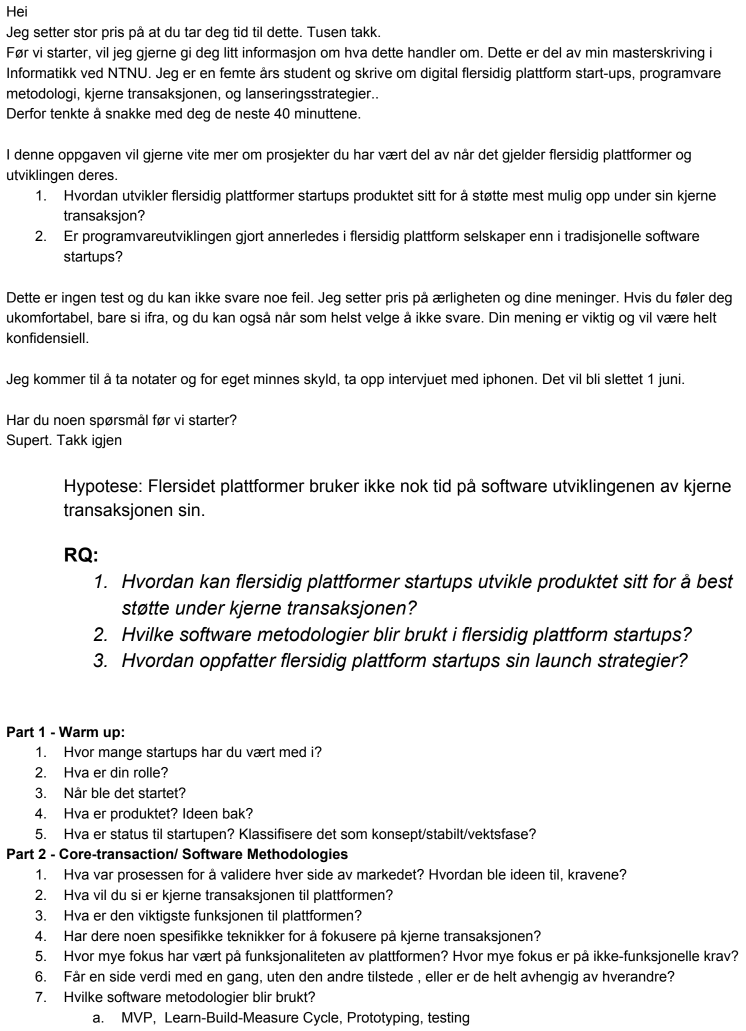
1. *How do multi-sided platform startups develop their product in order to best support their core transaction?*
2. *How is software engineering done currently in multi-sided platform startups? (Which software methodologies are most important?)*
3. *How do multi-sided platform startups perceive and manage launch their launch strategies?*

Part 1 - Warm up:

1. What is the product?
2. How many multiple-sided platforms have you been a part of?
3. What was your role in these companies?
4. When was it founded?
5. What is the current stage of the company? Concept? Growth? Stabilization?

Part 2 - Core-transaction/ Software Methodologies

Figure 5.2: Interview Guide in English



Hei
Jeg setter stor pris på at du tar deg tid til dette. Tusen takk.
Før vi starter, vil jeg gjerne gi deg litt informasjon om hva dette handler om. Dette er del av min masterskriving i Informatikk ved NTNU. Jeg er en femte års student og skrive om digital flersidig plattform start-ups, programvare metodologi, kjerne transaksjonen, og lanseringsstrategier..
Derfor tenkte å snakke med deg de neste 40 minuttene.

I denne oppgaven vil gjerne vite mer om prosjekter du har vært del av når det gjelder flersidig plattformer og utviklingen deres.

1. Hvordan utvikler flersidig plattformer startups produktet sitt for å støtte mest mulig opp under sin kjerne transaksjon?
2. Er programvareutviklingen gjort annerledes i flersidig plattform selskaper enn i tradisjonelle software startups?

Dette er ingen test og du kan ikke svare noe feil. Jeg setter pris på ærligheten og dine meninger. Hvis du føler deg ukomfortabel, bare si ifra, og du kan også når som helst velge å ikke svare. Din mening er viktig og vil være helt konfidensiell.

Jeg kommer til å ta notater og for eget minnes skyld, ta opp intervjuet med iphonen. Det vil bli slettet 1 juni.

Har du noen spørsmål før vi starter?
Supert. Takk igjen

Hypotese: Flersidet plattformer bruker ikke nok tid på software utviklingenen av kjerne transaksjonen sin.

RQ:

1. *Hvordan kan flersidig plattformer startups utvikle produktet sitt for å best støtte under kjerne transaksjonen?*
2. *Hvilke software metodologier blir brukt i flersidig plattform startups?*
3. *Hvordan oppfatter flersidig plattform startups sin launch strategier?*

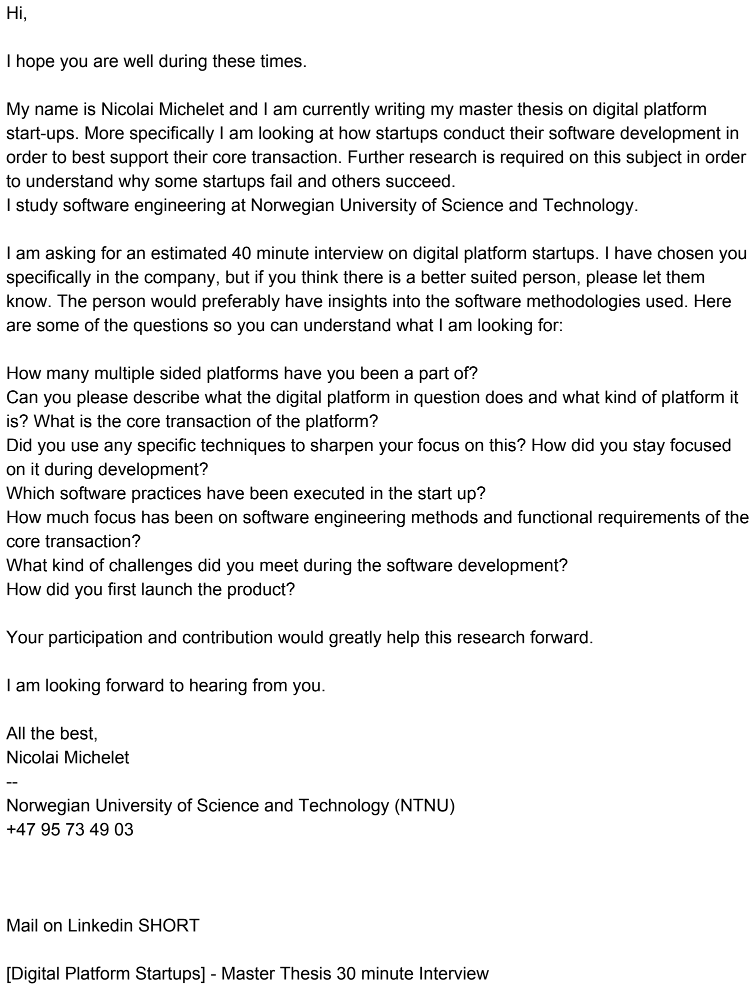
Part 1 - Warm up:

1. Hvor mange startups har du vært med i?
2. Hva er din rolle?
3. Når ble det startet?
4. Hva er produktet? Ideen bak?
5. Hva er status til startupen? Klassifisere det som konsept/stabilt/vektsfase?

Part 2 - Core-transaction/ Software Methodologies

1. Hva var prosessen for å validere hver side av markedet? Hvordan ble ideen til, kravene?
2. Hva vil du si er kjerne transaksjonen til plattformen?
3. Hva er den viktigste funksjonen til plattformen?
4. Har dere noen spesifikke teknikker for å fokusere på kjerne transaksjonen?
5. Hvor mye fokus har vært på funksjonaliteten av plattformen? Hvor mye fokus er på ikke-funksjonelle krav?
6. Får en side verdi med en gang, uten den andre tilstede , eller er de helt avhengig av hverandre?
7. Hvilke software metodologier blir brukt?
    a. MVP, Learn-Build-Measure Cycle, Prototyping, testing

Figure 5.3: Interview Guide in Norwegian



Hi,

I hope you are well during these times.

My name is Nicolai Michelet and I am currently writing my master thesis on digital platform start-ups. More specifically I am looking at how startups conduct their software development in order to best support their core transaction. Further research is required on this subject in order to understand why some startups fail and others succeed.
I study software engineering at Norwegian University of Science and Technology.

I am asking for an estimated 40 minute interview on digital platform startups. I have chosen you specifically in the company, but if you think there is a better suited person, please let them know. The person would preferably have insights into the software methodologies used. Here are some of the questions so you can understand what I am looking for:

How many multiple sided platforms have you been a part of?
Can you please describe what the digital platform in question does and what kind of platform it is? What is the core transaction of the platform?
Did you use any specific techniques to sharpen your focus on this? How did you stay focused on it during development?
Which software practices have been executed in the start up?
How much focus has been on software engineering methods and functional requirements of the core transaction?
What kind of challenges did you meet during the software development?
How did you first launch the product?

Your participation and contribution would greatly help this research forward.

I am looking forward to hearing from you.

All the best,
Nicolai Michelet
--
Norwegian University of Science and Technology (NTNU)
+47 95 73 49 03

Mail on Linkedin SHORT

[Digital Platform Startups] - Master Thesis 30 minute Interview

Figure 5.4: General mail to companies