CUSTOMER DISCOVERY AND CUSTOMER VALIDATION IN LEAN SOFTWARE STARTUPS

Tuomas Tähti

Master of Science in Technology Thesis

Software Engineering

December 2013

DEPARTMENT OF INFORMATION TECHNOLOGY

UNIVERSITY OF TURKU

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin OriginalityCheck service.

Turun yliopiston laatujärjestelmän mukaisesti tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -järjestelmällä.

UNIVERSITY OF TURKU
Department of Information Technology, Faculty of Mathematics and Natural Sciences

TÄHTI, TUOMAS: Customer Discovery and Customer Validation in Lean Software Startups

Master of Science Thesis, 77 pages
Software Engineering
December 2013

The traditional business models and the traditionally successful development methods that have been distinctive to the industrial era, do not satisfy the needs of modern IT companies. Due to the rapid nature of IT markets, the uncertainty of new innovations' success and the overwhelming competition with established companies, startups need to make quick decisions and eliminate wasted resources more effectively than ever before. There is a need for an empirical basis on which to build business models, as well as evaluate the presumptions regarding value and profit.

Less than ten years ago, the Lean software development principles and practices became widely well-known in the academic circles. Those practices help startup entrepreneurs to validate their learning, test their assumptions and be more and more dynamical and flexible.

What is special about today's software startups is that they are increasingly individual. There are quantitative research studies available regarding the details of Lean startups. Broad research with hundreds of companies presented in a few charts is informative, but a detailed study of fewer examples gives an insight to the way software entrepreneurs see Lean startup philosophy and how they describe it in their own words.

This thesis focuses on Lean software startups' early phases, namely Customer Discovery (discovering a valuable solution to a real problem) and Customer Validation (being in a good market with a product which satisfies that market). The thesis first offers a sufficiently compact insight into the Lean software startup concept to a reader who is not previously familiar with the term. The Lean startup philosophy is then put into a real-life test, based on interviews with four Finnish Lean software startup entrepreneurs. The interviews reveal 1) whether the Lean startup philosophy is actually valuable for them, 2) how can the theory be practically implemented in real life and 3) does theoretical Lean startup knowledge compensate a lack of entrepreneurship experience.

A reader gets familiar with the key elements and tools of Lean startups, as well as their mutual connections. The thesis explains why Lean startups waste less time and money than many other startups. The thesis, especially its research sections, aims at providing data and analysis simultaneously.

Keywords: Lean software startup, software entrepreneurship, Lean software development, Agile development

TURUN YLIOPISTO
Informaatioteknologian laitos, matemaattis-luonnontieteellinen tiedekunta

TÄHTI, TUOMAS: Asiakkaiden etsintä ja validointi Lean-ohjelmistokasvuyrityksissä

Diplomityö, 77 sivua
Ohjelmistotuotanto
Joulukuu 2013

Perinteiset liiketoimintamallit ja aiemmin menestyksekkäät kehittämismenetelmät, jotka ovat olleet tunnusomaisia teolliselle aikakaudelle, eivät täytä nykyaikaisten IT-yritysten tarpeita. Johtuen IT-markkinoiden nopeasta luonteesta, uusien innovaatioiden menestymisen epävarmuudesta ja kilpailusta asemansa vakiinnuttaneiden yritysten kanssa, kasvuyritykset joutuvat tekemään nopeita päätöksiä ja vähentämään resurssien haaskaamista tehokkaammin kuin koskaan ennen. On olemassa tarve kokeelliselle perustalle, jonka varaan liiketoimintamalleja voidaan rakentaa ja jonka avulla arvoon ja liikevoittoon liittyviä oletuksia voidaan arvioida.

Alle kymmenen vuotta sitten Lean-ohjelmistokehittämisperiaatteet ja –käytännöt tulivat akateemisissa piireissä laajalti tunnetuiksi. Nuo käytännöt auttavat kasvuyritysten perustajia validoimaan oppimisensa, testaamaan oletuksensa ja olemaan entistä dynaamisempia ja joustavampia.

Tämän päivän kasvuyrityksille on ominaista se, että ne ovat aiempaa yksilöllisempiä. Lean-kasvuyritysten yksityiskohdista on olemassa määrällisiä tutkimuksia. Laajat tutkimukset, joissa satoja yrityksiä esitellään muutamalla kaaviolla ovat informatiivisia, mutta yksityiskohtainen harvempiin esimerkkeihin perustuva tutkimus antaa syvällisempää tietoa siitä, miten ohjelmistoyrittäjät näkevät Lean-kasvuyritysfilosofian ja miten he kuvailevat sitä omin sanoin.

Tämä tutkielma keskittyy Lean-ohjelmistokasvuyritysten varhaisiin vaiheisiin, nimittäin asiakkaiden etsintään (engl. *Customer Discovery*, tarkoittaa arvokkaan ratkaisun löytämistä todelliseen ongelmaan) ja asiakkaiden validointiin (engl. *Customer Validation*, tarkoittaa hyvällä markkina-alueella olemista sellaisen tuotteen kanssa joka pystyy tyydyttämään kyseisen markkinan). Tutkielma tarjoaa ensin riittävän tiiviin katsauksen Lean-ohejlmistokasvuyrityskonseptiin sellaisille lukijoille, joille kyseinen termi ei ole entuudestaan tuttu. Tämän jälkeen filosofia asetetaan todelliseen testiin, joka toteutetaan haastattelemalla neljää suomalaista Lean-ohjelmistokasvuyrittäjää. Haastattelut paljastavat 1) onko Lean-ohjelmistokasvuyritysfilosofia aidosti arvokas yrittäjille, 2) kuinka teoriaa voidaan käytännöllisesti soveltaa ja 3) korvaako teoreettinen Lean-kasvuyritystietämys puuttuvaa yrittäjyyskokemusta.

Lukija perehdytetään Lean-kasvuyritysten keskeisiin elementteihin ja työkaluihin sekä näiden keskinäisiin yhteyksiin. Tutkielma selittää, miksi Lean-kasvuyritykset haaskaavat vähemmän aikaa ja rahaa kuin monet muut kasvuyritykset. Etenkin tutkielman tutkimusosiot pyrkivät esittelemään asiatietoa ja analyysiä rinnakkain.

Asiasanat: Lean-ohjelmistokasvuyritys, ohjelmistoyrittäjyys, Lean-ohjelmistokehittäminen, Agile-kehittäminen

Table of Contents

Figures and Tables

1 Introduction

A startup is a company which is expected to grow fast. Just being newly founded does not make any company a startup – expecting growth is essential by definition. [1]

Most startups fail. This thesis is about software startups, and in the software business success is no easier achievement than elsewhere. Software entrepreneurship is a rather young concept in both scientific and business sense. It should be expected that there is uncertainty regarding its best practices and principles. Due to the unstable and unpredictable nature of the markets and the competition among companies, software startups need to test their assumptions and learn fast in order to survive and grow.

One way for a software startup to learn fast and stay flexible is to adapt Lean software development principles. The term *Lean software development* originates in a book by that name, written by Mary and Tom Poppendieck (2003). [2] The principles were not invented by them from a scratch, but their book explains how the principles from physical manufacturing (Toyota Motor Corporation) offer a valuable approach to software development. In the software world Lean development is also linked to the *Agile Manifesto* (2001) which aims to emphasize lightweight development methods that result in constant progress pace, easy-to-change plans and fast product delivery.

In Lean thinking, today's business plan is considered better than yesterday's plan by default, for today a startup has more information to support its plan. Traditionally many companies have stoutly followed their old plans in many kinds of situations. Lean principles do not aim at replacing the old methods, but to supplement them. Different companies, who are in different points of their lifespan, need different methods.

The growth of a successful startup usually has three phases:

1) First there is a time of slow or no growth, while the startup figures out what it should be doing.

2) When the startup realizes how it can make something that many people want and how to reach those people, and how to do all this profitably, there's a period of more rapid growth.

3) Later on the startup scales into a big company. Growth will slow down when the company reaches the limits of the market it serves. [1]

This thesis focuses on Lean software startups' early phases prior to scaling. These phases are Customer Discovery (discovering a valuable solution to a real problem) and Customer Validation (being in a good market with a product which satisfies that market). The thesis first takes a look at the Lean principles and the search for a profitable way of doing business, and then discusses a few Lean startup case studies.

The thesis also aims to offer a sufficiently compact insight into the Lean software startup concept to a reader who is not previously familiar with the term.

Chapter 2 provides the reader with historical background information about Lean thinking, what has made Lean special and what Lean is not. The scope of this chapter is broad in the sense that the content is not limited to software companies.

Chapter 3 introduces the key elements and tools of Lean startups, as well as their mutual connections. By doing this, the chapter explains why Lean startups waste less time and money than many other startups, and how Lean startups achieve rapid progress and validated learning.

Chapter 4 discusses a highly practical and crucial issue: metrics and tests. In the search for a good way of doing software business, testing and optimizing are always immanent. This chapter takes a look at what should be tested, and how. Any startup will run out of resources if it tries to observe and test everything without prioritization.

Chapter 5 puts the tools and ideas into a real-life test. The chapter is based on interviews with four Finnish Lean software startup entrepreneurs. Those interviews reveal 1) whether the Lean startup philosophy is actually valuable for them, 2) how can the theory be practically implemented in real life and 3) does theoretical Lean startup knowledge compensate a lack of entrepreneurship experience.

Finally, Chapter 6 provides the reader with a summary and conclusions.

2 Lean and Agile Thinking

2.1 History

Agile and Lean are two keywords being widely used nowadays in most of corporate world engaged in production or development. The Lean and Agile concepts are connected: Lean centers on reducing all kinds of waste as much as possible, while Agile methods are used for fast introducing new products in a changing market which is characterized by unpredictability. The ability of responding to a changing environment requires innovative staff members and an adaptable organizational structure. [3] This chapter discusses both Agile and Lean.

2.1.1 Manufacturing: Toyota Production System

The manufacturing methods of Toyota Motor Corporation are historically a major precursor of Lean thinking. The Toyota Production System (TPS) was centered on a few main pillars: continuous improvement (often called by the Japanese word *kaizen*), respect for people, teamwork of multi-operational workers and setting optimal profit margins.

Continuous improvement cannot be maintained in a company where managers and employees consider the circumstances to be completely satisfying or even good enough. Toyota's approach to doing business is to challenge all conditions and assumptions in order to locate improvable processes. These improvements are not the only valuable outcome of the *kaizen* concept. Another gain of this thinking is creating a company atmosphere of welcoming changes and learning constantly.

The second pillar, respect for people, is necessary because continuous improvement is impossible if people's fresh ideas and opinions are not taken seriously. Employees should be boldly encouraged to speak up their minds. Toyota has "actualized this

respect by providing employment security and seeking to engage team members through active participation in improving their jobs". [4] The goal is to advance the creative mentality of an individual worker, but TPS focuses also on the mentality of teamwork.

The third area of concern is creating a mentality of teamwork. The TPS perspective of team's work flow is analogical to relay running where baton is passed from one athlete to another. As soon as a worker finishes processing a part, he passes it on to the next worker. If that worker is delayed for some reason, other workers who may be available in the sector should help him. Making the slowest process faster makes the whole overall process faster.

To enhance work flows, leaders of Toyota Corporation have decided that instead of putting all identical machines – i.e. machines of the same process – into one physical location in the factory and carrying parts back and forth between processes, machines should be placed according to the work flow order. This way it was also possible to assign one worker to operate more than one machine: the policy of "one operator, many processes" was born. This practice increased production efficiency 2–3 times, compared to "one operator, one process". [5]

Another primary principle of the Toyota Production System is setting the optimal profit margins. Pricing most often means balancing between satisfactory profit and what customers find reasonable. If the product is not absolutely necessary, the potential customer will not buy it when price is too high. Most people can live without a car, so Toyota drew the conclusion that in their case it is the customers, along with competitors, who set the price. That is why Toyota's formula for profit is

```
selling price – actual cost = profit
```

In this formula selling price is not a variable that the company can freely choose. Profit is incremented through cost reduction, not by increasing selling price. To reduce costs, Toyota instituted production leveling. For example, if some part is needed at the rate of 1000 units per 30 days, it is enough to make 40 parts a day for 25 days. Furthermore, if each workday consists of 480 minutes per workday, one part should be made every 12 minutes, and producing any more would just create overstock. This is not desirable, so the focus is on maintaining a reliable and steady supply of raw materials.

One thing that encourages people all around the world to explore TPS is that the corporation has very openly shared its intellectual source of competitive advantage. As early as in 1982 the Chairperson Eiji Toyoda and President Shoichiro Toyoda agreed to create a car manufacturing venture together with General Motors, with the explicit intention to teach TPS to General Motors, one of Toyota's global competitors. Ten years later in 1992 the Toyota Supplier Support Center (TSSC) was founded to support American Non-Toyota suppliers with their understanding of the TPS. According to TSSC, its mission is to "contribute to society by sharing TPS know-how and improving the general level of public institutions and North American manufacturing industry". [6]

Familiarizing oneself with the history of TPS is recommendable and educational, but it is not enough, for TPS is still evolving in today's Toyota. TPS was created largely because of the urge to meet the needs of Japanese consumers who had very limited amount of capital compared to today's situation. Every company must design its own, more or less unique way of doing business. The "Toyota Way" is a special concept that should not be – and most probably cannot be – blindly copied to any other organization without making modifications. This is one of the reasons why Toyota has not kept TPS as a secret.

2.1.2 Software: Agile Manifesto

The term *Agile software development* means a cluster of software development methodologies which are built on the idea of iterative working. The teams should be self-organizing and cross-functional.

The public foundation of Agile software development is the *Manifesto for Agile Software Development*, also known as the *Agile Manifesto*. The manifesto was originally signed by 17 software developers and published in February 2001. The aim of the manifesto was to emphasize lightweight development methods that result in constant progress pace, easy-to-change plans and fast product delivery. The focus is on four choices: [2]

1) To value individuals and interactions over processes and tools

When professional developers are involved, teams ought to be self-organizing and willing to use state-of-the art interaction methods, e.g. pair programming.

2) To value working software over comprehensive documentation

Delivering working software as soon as possible and listening to the feedback gives a valuable sign of whether the project is heading to the right direction or not; documentation is important only if the software works.

3) To value customer collaboration over contract negotiation

Requirements cannot be completely realized in the beginning of the work cycle, whereupon continuous customer/stakeholder involvement is valuable.

4) To value responding to change over following a plan

Traditional development models are often like cannonballs: once a plan is launched, changing its direction is nearly impossible. This works if and only if the customer knows what he wants and nothing changes along the way. Compared to a cannonball, Agile is like a homing missile that can deal with changing requirements. The assumption here is that plans are basically nothing more than guesses. There are usually many factors that cannot be controlled: market conditions, competitors, customers and so on. Long-term plans let the past drive the future and neglect the fact that knowledge increases as any project goes on; one has the most information when he's doing something, not before doing it. [7]

2.2 Agile Triangle – Setting the Stage for Measuring Success

As mentioned in Section 2.1, Agile values responding to change over following a plan. Changes might result in situations that seem like failures, such as spending money over the initial budget. But Agile is flexible: constraints are kept in mind, but the product's current value and potential future value are also very important. This gives a comprehensive viewpoint for measuring success.

The classical schoolbook way to measure the success of a profit-seeking project, be it software-related or not, is the Project Triangle (a.k.a. Iron Triangle) which is shown in Figure 1. The tips of the triangle represent cost, schedule and scope. It is normal that a customer wishes to lock all dimensions on day one of the production. If two or three sides of the triangle grow significantly longer than expected, a project might be considered a failure.

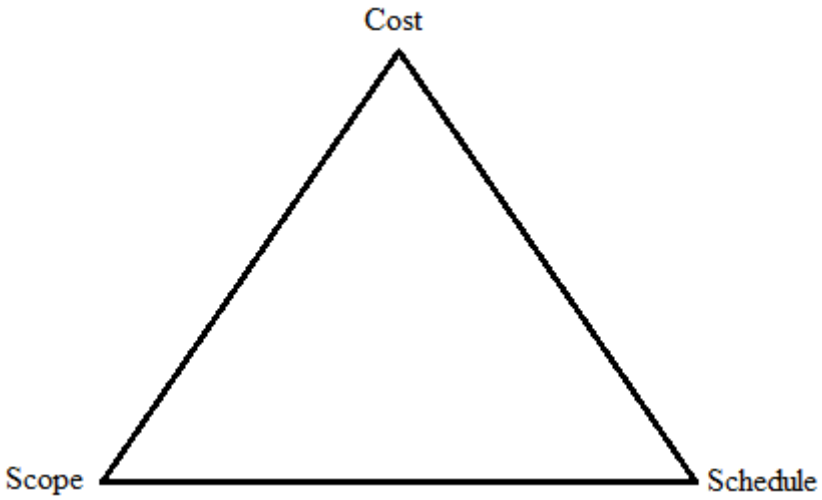

Figure 1. Project Triangle.

Let us measure the success of the 1997 movie *Titanic* in respect to the Project Triangle. The filming was expected to last 138 days but eventually it took 160 days due to the director's ambitious plans. The premiere date was scheduled for July 1997, but was postponed by five months. The production cost $200 million – much more than the initial budget, making *Titanic* the world's most expensive film at that time. [8]

In other words, *Titanic* was a failure. Except that the movie's global gross is more than $2 billion and it was the first cinema to ever reach the one-billion-dollar mark. Furthermore, the film was critically acclaimed and received eleven Academic Awards.

What happened with *Titanic*, can analogically happen in software projects. The Project Triangle only focuses on the constraints, ignoring the work's actual goal which is value – both the product's current value and its potential future value. These are notified as two tips of the Agile Triangle, whereas constraints are the third point, as seen in Figure

2. The goal of any development project should be to optimize value within given constraints.

Many sources mention *quality* instead of *potential future value* as the triangle's third tip. However, quality is just one means for value and quality has an impact on both the current and the future value. Current quality refers to the reliability of the latest product version. When a product operates in a reliable way, it has value in the form of implemented features. Future quality means the long-lasting ability to continuously deliver value. Responding to changes in the outside world is as easy as expanding and updating the code. All code should be written with decent practices and proper documentation. Keeping the "technical debt" low increases the adaptability. [9]

Agile teams should focus on the releasable product's value rather than being limited by the Project Triangle. If a company is building something that nobody wants, it does not matter much if activities are being done on time and on budget. Once the product is released, it is the value what matters to the stakeholders.

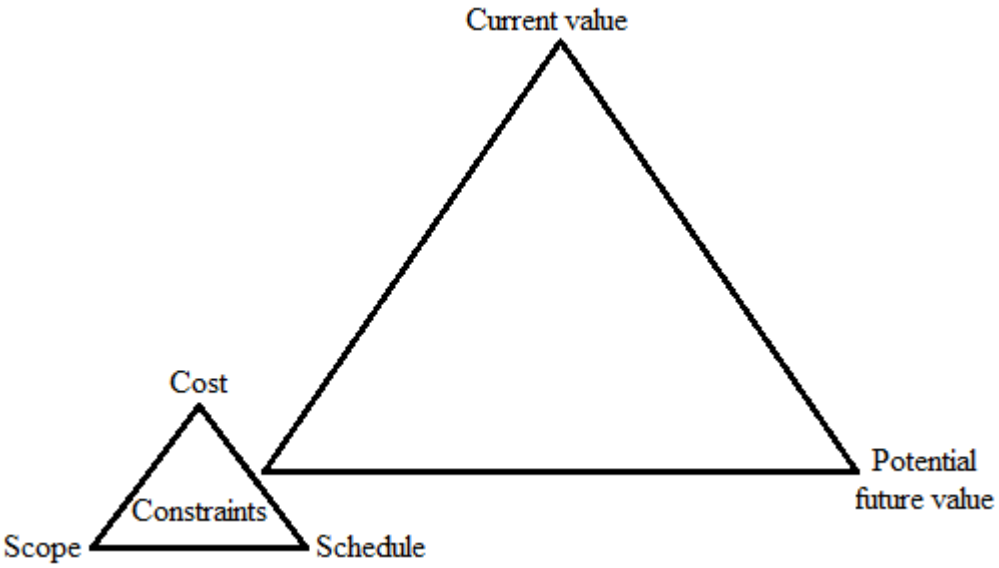

Figure 2. Agile Triangle.

A feature or function of software has no value if it is not useful to the customer. In 2002 Standish Group released a 2000-project-covering statistic which showed that 45% of features and functions were never utilized by a typical user, meaning that they are waste

(Section 2.3.2). [10] Unnecessary features postpone the release and cost money for the developing company. Rather than asking “Have we implemented everything that was required?” the developers ought to be thinking “Is it already possible to release the product?” Getting user feedback as early as possible is one of the key ideas of Agile development.

2.3 Lean Software Development

2.3.1 Difference between Lean and Agile

In today’s markets, Lean and Agile are the two most popular theories of developing products quickly, efficiently and at the lowest possible cost. Lean emerged longer ago; Agile is relatively new and includes certain features from Lean, but there are also differences between the two concepts.

Lean provides an overall way to make a process more effective. The Lean philosophy can be implemented in any industry and any business type. Simply put, Lean aims at reducing all kinds of waste in the work processes.

Those who follow the Lean practice are proactively identifying their opportunities to improve product quality. By asking for early feedback they can quickly react to development mistakes. The greatest advantage of Lean is continuous learning and improvement.

Agile is a relatively recent concept that aims to take the best features of Lean and add some new ones. While Lean is a *philosophy* of a wide scope, Agile is more of a *technique* that is precisely suitable for fast software development in an unpredictable environment. [11] Agile means a strategy for introducing new products in a continuously changing environment of unforeseen occurrences. Agile strategy focuses on how operations can respond to a changing environment. It is the ability of a company to be open to opportunities and be prepared to effect changes.

Agile gets products to the market fast by reducing the development cycle. Failing projects are cancelled quickly, and thus major losses are avoided. Agile reduces costs by failing faster, whilst Lean believes in frugality in the development process and the focus of Lean philosophy is on reducing waste as much as possible. [3]

2.3.2 Lean Software Development Principles

The principles and tools of Lean software development (Lean SWD) are process-centric, primarily aiming to improve the work instead of workers. Much of the Lean thinking can be encapsulated to the proverb "people do not fail, processes do". Even when someone makes a personal mistake, the reaction should be analyzing why did the tools and practices make that mistake so easy to be made.

Software developers might use their hours ineffectively or have issues with mutual communication. Another typical case is that after a system is considered ready to be used, it takes weeks or months to deploy it. These situations can look like problems as such, but they are nothing but symptoms of the underlying causes. Symptoms can also be something that are not present in the development process, such as unsatisfied customers.

It easily happens that people recognize what is wrong in the system, but fail to see the fundamental reason behind it. Lean thinking advices to keep asking *why* until one gets to the root of the problem. This method is generally known as *Five Whys*, although the number of steps is not fixed. The method encourages teams to diagnose issues in their processes and to work for preventing the risk of teams getting slower over time. In the ideal case, teams are able to accelerate even when they scale their activities. Each question takes thinking deeper to the origins of the problem whose existence was first noticed only because of its consequences. [12] [13]

This thesis concentrates on information that is relevant to software startups, but Lean SWD principles are by no means limited to newly founded growth-seeking companies. In startups, some principles require a lot of attention and discipline, while some others might be followed without the staff even noticing it.

The seven principles that summarize Lean development are: [2]

• Eliminate waste

• Amplify learning

• Decide as late as possible

• Deliver as fast as possible

• Empower the team

• Build integrity in

• See the whole

The continuous improvement of processes aims at **eliminating waste**. Waste means any activity that does not add value to the product, i.e. anything the customer is not willing to pay for. Some value-added activities, such as building desired features and shipping the product to customer, are intuitively figurable. In software projects, waste includes unnecessary coding, waiting/delays, unclear requirements, inadequate testing and slow communication.

The standard way to think about waste elements is the timeline from entering order to receiving payment, as seen in Figure 3. The timeline consists of several activities in chronological order. Eliminating waste from any part of the whole process makes the timeline shorter.

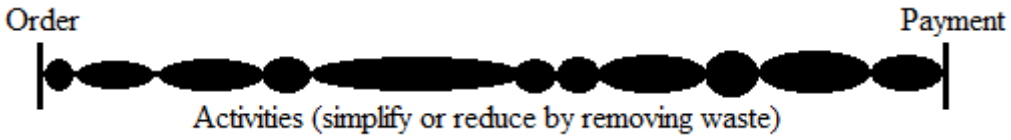

Figure 3. Timeline of activities from product order to payment.

Eliminating waste improves the processes of software development, but also the development environment should be constantly improved, meaning that **learning is amplified**. Speeding up the learning can be achieved by implementing short iteration cycles, which include feedback sessions with customer representatives. Learning

happens on both sides of the table: development team hears how to fulfill the customer's wish, and the customer gets better understanding about what should be done next and what he really needs. In Chapter 3 it will be shown that this learning is crucial for software startups.

Uncertainty is always immanent in life, and especially in startups. However, the more time goes by the more one can learn about the surrounding world. This is why **decisions should be made as late as possible** – they should be based on facts rather than assumptions. This means continuous planning combined with the readiness to abandon yesterday's plan. Lean software development principles can bring forward the creation of different options and delay some important decisions until the time when customers have understood their needs.

Although Lean principles tell to decide as late as possible, developers should **deliver as fast as possible**. This serves the will to amplify learning: the sooner a reasonably functional version of the end product is delivered, the sooner some valuable feedback can be heard. *Any* clear and concrete feedback potentially eliminates waste from the next iteration cycle.

Teams are empowered by avoiding micro-managing. Lean software development teams are sovereign and flexible all the way to being self-organizing and having direct access to the customer. When the customer gives input concerning his current needs, the developers divide the tasks in a way they find reasonable, and estimate the time that is necessary for each task. Then, every morning during a stand-up meeting each member of the developer team reports what he did yesterday, what he'll do next and what information he needs from the others. Of course this requires that company managers trust the developers to do their job in an unprompted manner. For a small startup consisting of a handful of people who all know each other, this should not be a problem.

Built-in integrity means that the system's components work well with each other and there is balance between maintainability, efficiency and responsiveness. This means that the customer's problems and needs must be seen as a whole. Once again good internal communication and transparency in the organization plays the key role. Furthermore, the information flux should be going uninterruptedly from customers to developers and back, in order to prevent developing in isolation.

Software products are not just puzzles that are put together from hundreds of individual pieces. Software is all about the interaction between those pieces. Modern software systems are the most complex entities mankind has ever created, so good integrity is not always easy to reach. Developers need to understand all previous Lean principles and most of all they need to **see the whole**. Team members must be interested in the final output, for their individual work is just one part of it.

The heart and spirit of Lean principles is not to avoid mistakes. That would be unrealistic. The collective goal of Lean principles is to provide the entrepreneur with what he needs when he fails: when he takes a bad turn, he has the necessary tools to realize it, and the agility to find a better path.

2.4 What Lean is Not

Any innovation can be misunderstood. The value of a tool or method is revealed only in proper usage. Lean has sometimes been mistaken as a solution to certain problems that it wasn't created for. It should be clarified that Lean is not a magic bullet that makes startups flourish.

Although Lean underlines the importance of reducing resource losses and eliminating waste, applying Lean in a startup is not necessarily cheap. Lean is meant for increasing productivity and minimizing the time spent, but not for minimizing the money spent. Product development cycles and validated learning help entrepreneurs turn hypotheses into facts and leave the false assumptions behind. As a result, financial capital may be saved in the long run, but spending as little money as possible in the startup phase is not an objective as such.

Lean methodology is not a formal standard that one can mechanically implement. Lean processes may be used to help a startup meet official standards and legislative requirements, but this takes creativity from the entrepreneur's side. Lean is a set of tools, not a ready-made product. [14]

Another myth – or rather an exaggeration – states that Lean thinking replaces all visions with non-negotiable data, such as customer feedback. This false impression occurs when one or more assumptions of the startup idea are discarded in the process of changing the business plans. Changing the course because something new is learned can seem like giving up the original product vision, but its purpose is to make the product realistically profitable and find the right vision rather than reject being visionary. [15] The obvious problem here is that when someone tries to validate his hypothesis, he can easily delude himself into perceiving it as correct.

Most startups fail. Whether or not Lean is implemented, there is a significant amount of uncertainty in the startup's progress, for in the beginning both the product and the customer segment are unknown. Startup teams should not trust Lean tools blindly and ignore everything else, because eventually it is the team itself that makes the difference between a success story and a failure. Fast iteration cycles and comprehensive customer enquiries are only worth so much if the team is unable to draw the correct conclusions and change the course when necessary. The good news is that in Lean learning can be validated, whereupon failing once does not mean that the team shall fail again. In the world of traditional product development failing has been a somewhat shameful outcome. In Lean IT startups failing means losing time, effort and money, but it does not predict the future of the team's next startup.

3 Lean Startup

By definition, a startup is a company expected to grow fast. If you open a barber's shop and if you do not plan to hire any workers other than yourself, you have not created a startup because you will not be scaling your business. The process of scaling a startup into a big company is outside the scope of this thesis, but the idea that the startup is *designed* to grow is essential.

Another difference between a barber's shop and a typical software startup is that a barber can expect the business environment to stay somewhat stable and predictable. People will always want haircuts and human hair grows at a constant rate. Software startups should assume that the surrounding business ecosystem is both unstable and unpredictable. This is the environment where Lean tools are most useful.

The Lean startup method is all about testing the assumptions and doing it fast. Lean startups waste less money than other startups because they have a planned approach to testing new ideas. Speed is achieved through iterative processes, validating what has been learned, pivoting when necessary and getting customer feedback early. Learning happens in iterative loops.

Lean startup is a term coined by Eric Ries. It is built on the management processes Ries employed while he was the Chief Technology Officer at IMVU, a highly successful social entertainment online community. [16] This chapter introduces the most important key concepts and then discusses how the way of doing business is searched during the early phases of a software startup. The concepts are coherent with the Lean SWD principles of Section 2.3.2.

This chapter mentions various teams, such as product development team and pre-sales team, as parts of the startup personnel. The word *team* is not to be taken literally, for early startups may consist of 3–5 persons or so. Teams refer to individuals or small groups who have certain enduring responsibilities.

3.1 Key Concepts

3.1.1 Minimum Viable Product

A Minimum Viable Product (MVP) has just those features that allow the software to be introduced, and nothing more. It may be an entire product or a sub-set, such as a feature. The idea of quickly releasing MVPs that obviously lack many features of the actual release is to test the assumptions and minimize the risk and loss you face on the route to discovering a proper business plan. Developers want to know what is the smallest set of features a customer is willing to try in the first release.

The biggest mistake one can do concerning MVPs is to add needlessly many features. Technically such a product is not even an MVP. As an example, one fast way to test the assumptions for an application is to make a mock website for the product and buy advertising to get traffic to that site. The number of clicks on the project's landing page indicates the interest towards the forthcoming product.

A link to a single new feature of a web application may be placed in a visible location on the existing product website. The feature is not yet implemented; it is only described with a link to a page that says "This feature is currently under development, check back soon". Clicks are again recorded and they provide an indication to the demand for such a feature in the product's existing customer base.

In Lean software development one product has multiple MVPs as the development goes on and new assumptions are tested. Building the next-phase MVP that is more than just a landing page can be done with relatively low costs, thanks to the current tools. For example, connecting the new web application to some social media service saves the time of creating membership database, the help service for lost passwords and so on. [17]

When talking about software MVPs that are actual programs and not just landing websites, the common pattern is that the time between two MVP versions gets shorter as

time goes by. For example the time from starting the development until the very first MVP could be six months, then four more months until the second MVP, then only two months until the third MVP is ready, and so on. The reasons for this progress are that developers learn more about their assumptions and they do not have to start from a scratch. [18]

Minimum Viable Products should draw the attention of customers, but they are not tools for PR. The first MVPs may have major downsides and look almost unrecognizable compared to the actual product. Using financial capital for promoting an MVP might not only be waste of money, but also a risk of public embarrassment.

3.1.2 Validated Learning

Releasing new products and making money are not the only goals of Lean software startups. Startups are also ways of learning how to build sustainable business. In order to make sure that the learning is real and that it focuses on the right things, it must be empirically validated.

A startup collects its validated learning little by little, and validated learning can be seen as a unit of progress. [19] Validated learning is achieved through testing. That is to say, the hypotheses and visions of the entrepreneurs are continuously challenged – a logical action in an environment of omnipresent uncertainty. The startup team wants to know if they are making something actually valuable, or are they just playing "success theater" where meaningless metrics show good results and meaningful metrics are ignored.

Individual startups are utilizing somewhat different metrics to measure their success. Whatever the chosen metrics are, the essence of validated learning is always the same: it is demonstrated by improvements in the startup's core metrics. No estimations and no guesses; just undisputable empirical information which is received directly from the customers.

The reality is not that simple, though. Any software startup begins with a vision, and the goal of experiments is to figure out how to create a sustainable and scalable business model around that vision.

Let us say that the customers are giving conflicting statements about the product and its features. Which customer opinions should be considered important? What is the best way to prioritize the future changes, and what can be changed safely?

In the Lean Startup methodology the work being done is considered an experiment, and the test subjects are the entrepreneurs' assumptions. In the first place, entrepreneurs must have clear hypotheses and written expectations about how the product should succeed and what is expected to happen in overall. The predictions are tested empirically by carrying out the plans and retrospectively checking how well the expectations were met. This happens in short cycles of the Learning Loop (Section 3.1.5).

The entrepreneurs might innovate something completely new that customers have not thought before. Therefore the customers might not know what they really want, until something like that is explicitly presented to them. At best, Lean development of a product is a process where the developers learn about customer needs and customers realize what is good for them.

Plans and hypotheses are all parts of the *business model*. This term shall be discussed in Section 3.2. For now it is enough to state that validated learning is a tool for testing the elements of a startup's business model.

It is recommendable for startups to aim at validating the learning even when the number of customers is still very low. At this stage quantitative research is pointless, but the entrepreneurs are more likely to know every customer rather well and keep contacted. With new product features and product marketing the entrepreneurs test the customer interest to gradually find the right formula for acquiring and satisfying the customers in the market segments that they targeted. They will also know how much it takes effort to get a new customer and how much revenue can be expected from the average customer.

3.1.3 Engine of Growth

It has been mentioned earlier that startups are designed to grow. Thus, a startup needs something that drives its growth in the sense of increasing the number of customers. This "something" is called the engine of growth. There are basically three engines of growth that power startups: the sticky, the viral and the paid. [18] [20]

The strategy in the sticky engine of growth is to keep the customers for a long time, whilst also gaining some new customers. The weekly or monthly number of new customers does not have to be big, if almost all of the old customers keep using the startup's product. The focus is on customer satisfaction and binding, not on marketing. In practice this means such things as online help service, product tips, user groups and active company blogs.

The viral engine of growth counts on grapevine and gossips. The idea is that a good product advertises itself when customers and potential future customers talk about it with each other. The trick is that building a good reputation for a product without actual advertising is very difficult to do intentionally. In everyday life it is more common for rumors to spread spontaneously and there is no guarantee whether the rumors are favorable or not – but the viral engine of growth has its success stories. Probably the best-known case is the web company Hotmail which started offering people e-mail accounts in 1996. In the beginning the company achieved only modest growth, but this changed when the owners made one simple modification to the system. They added into every sent e-mail the message "P.S. Get your free e-mail at Hotmail" along with a link. This tweak got the company over one million new customers in six months, and 12 million in 18 months. At this point Hotmail was sold to Microsoft for $400 million. [18]

Any kind of paid advertisement is a sign of the paid engine of growth. When operating on this engine, customers need to give the startup a profit that covers the expenses, and more preferably even surpass the ad costs. The gained money can be used for more advertising to accelerate growth.

The three engines of growth are not mutually exclusive, but for a startup it is best to focus on just one. If the entrepreneurs make their product sticky, manage to spread viral

information and buy new customers at the same time, it is difficult to figure out which engine is actually working. [20]

3.1.4 Pivot

Pivoting is the action of abandoning a hypothesis that proved to be wrong, and steering the startup to a new direction that has not been tried and tested yet. Pivot is the core of the "fail fast" concept. Startups go through multiple pivots that form a chain of changes. The business plan might go through remarkable remakes in the process, but this does not happen in one go. A singular pivot focuses on just one detail of the business plan, such as feature set or customer segment. Between pivots the entrepreneurs get an increasingly better understanding about what they should be doing, and at the time of each pivot there is more knowledge to support the decisions. [21]

The opposite of pivoting is to persevere and keep optimizing the product and business plan within the frame of old hypotheses. What is sometimes even more difficult than pivoting itself is to know *if* and *when* one should pivot. The question is "Are we making progress and creating something actually valuable, or should we change something in our assumptions?"

The choice to either pivot or persevere is unavoidably affected by human element. The downside of this is that there is no mathematically proved way to guarantee that making a pivot is a good choice. On the other hand, entrepreneurs are allowed to use their natural intuition, internal judgment and ability to see signals in noise. Pivots are supported with test results, but it is a human being who analyzes those results.

The more money, time, and other resources have been used for an idea, the harder it is to pivot and throw that idea away. For this reason it is essential to deliver MVPs as soon as possible. Another significant action to make pivoting easier is to explicitly identify all leap-of-faith assumptions in the startup.

Whenever a pivot is made, entrepreneurs should have numerical expectations of the results. If they just change something in the business plan and want to see what

happens, they always automatically succeed – in seeing what happens. Furthermore, the variables that are measured must be carefully chosen.

For instance, a startup may have a landing website for their upcoming product, and the website has a pre-order button. Let us say that initially the product was marketed in the business-to-business (B2B) manner, but later on the startup entrepreneurs decided that they will market the software for consumers (B2C). Switching from business-to-business to business-to-consumer is a clear pivot case. Let us assume that before pivoting the startup's website had a weekly average of 1 000 visitors and 2 pre-orders, and after pivoting those numbers increased to 8 000 hits and 4 pre-orders. Three conclusions can be made:

1) the number of visitors went up 700 % due to pivoting

2) the weekly amount of product pre-orders doubled from 2 to 4

3) before the pivot 0.2 % of site viewers wanted to buy the product, whereas after the pivot this number was only 0.05 %

Having more audience is encouraging, but it is the revenue what matters on payday. Chapters 4 and 5 pay more attention to the metrics that Lean software startup staff members are using.

When is the best time to decide whether to pivot or persevere? Since pivoting can be delayed for ambitious emotional reasons, it is recommendable for startups to have regular "pivot or persevere" meetings, for example once every month or once in two months. At minimum, the meeting needs participants from the developer team and the business team.

The necessity of pivot is not limited to cases where the startup is going down in terms of financial capital and the number of customers. One can make an MVP that successfully attracts the attention of early adopters and gives the startup a good start, but this does not guarantee the popularity among mainstream users who form the majority of all. In other words, there are different types of pivots. Here are short definitions for those types that are most relevant in the scope of this thesis.

Zoom-in Pivot

In this pivot type, what previously was considered just one feature among the others, becomes the whole product. The other features are dropped.

Zoom-out Pivot

It happens quite often that one single feature is not enough to support the product. In a zoom-out pivot, what used to be considered the whole product, becomes just one feature of a somewhat larger whole.

Customer Segment Pivot

A customer segment pivot happens when a company realizes that the product it is making does indeed solve a real problem for real customers, but those customers are not the kind of customers the company originally planned to serve. The product hypothesis is partially but only partially confirmed, solving an important problem but for a different customer group than what was originally assumed.

Customer Need Pivot

When getting to know customers well, it sometimes shows that the problem you are trying to solve is not that important after all. However, because of being close to the customer, you may discover more significant problems that you can solve. Again, the product hypothesis is just partially confirmed.

Platform Pivot

"A platform pivot means a change from an application to a platform or vice versa. Most commonly, startups that aspire to create a new platform begin life by selling a single application, the so-called killer app, for their platform. Only later does the platform emerge as a vehicle for third parties to leverage as a way to create their own related products." [18]

Business Architecture Pivot

"Companies generally follow one of two major business architectures: 'high margin, low volume' or 'low margin, high volume'. The former is commonly associated with business-to-business (B2B) or enterprise sales cycles, and the latter with consumer

products. In a business architecture pivot, a startup switches its architecture. Some companies change from ‘high margin, low volume’ by entering mass market, while others, originally designed for the mass market, turn out to require long and expensive sales cycles.” [18]

Value Capture Pivot

The monetization or revenue model is tweaked to capture value, with corresponding changes in business, product, and marketing strategies. Simply put, this pivot is a change in pricing models. It should be highlighted that monetization is not an isolated feature of a product that can be added or removed at will. Capturing value is an intrinsic part of the product hypothesis. Changing the way a startup captures value can have very far-reaching consequences.

Technology Pivot

A startup achieves the original solution using a different technology that can provide superior price and/or performance to improve competitive condition.

Engine of Growth Pivot

“In this type of pivot, a startup changes its growth strategy to seek faster and/or more profitable growth.” [18]

3.1.5 Learning Loop

Traditional development happens in a flux that starts with designing and building products based on relatively simple ideas. The process is product-focused and pays more attention to marketing than on customer interviews and learning. This strategy is still valid for many established companies and also for many new companies who do not seek growth. For startups, however, Lean methodology provides an alternative to the old waterfall plan: the Learning Loop.

The Learning Loop is the last of this chapter's key concepts for the reason that previously discussed terms – MVP, Validated Learning and pivot – are crucial elements of the loop.

The Learning Loop is illustrated in Figure 4. The starting point of the loop is to have a vision of a product that is assumed to be valuable for some group of customers. The loop is sometimes called "Build-Measure-Learn Loop", and while this is the direction of the loop, learning should actually come before building. [22] It would be waste to code something that nobody is interested in. Once the entrepreneur has a product idea, he should examine how customers react to it – and who the customers are. After gathering this data it is time to draw conclusions and either start over with an enhanced hypothesis or start building.

Even though the product idea is tested before building, many more iterations are still needed during the development. That is why "building" in this context means just creating the first MVP and launching it as soon as possible. However, quality of the code must never be sacrificed for the speed of coding. Low quality, as well as insufficient integration, causes cumulative problems and delays as the process goes on.

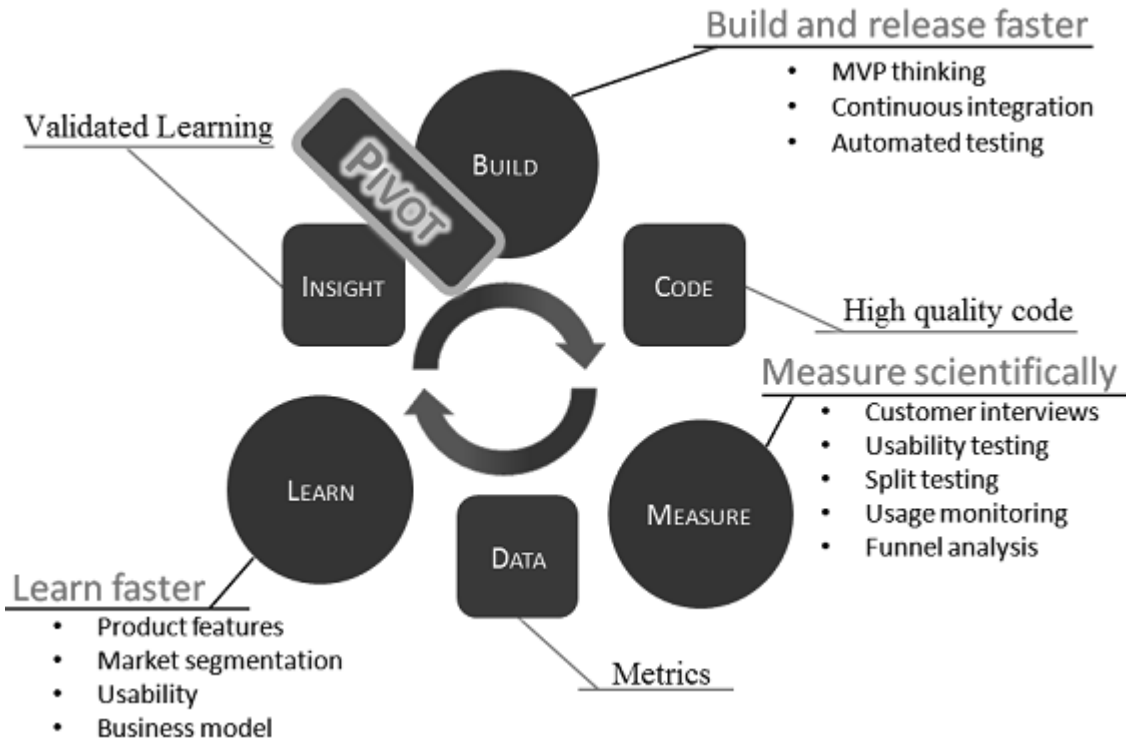

Figure 4. Learning Loop. Based on the work of Eric Ries. [18]

In the Measure part of the loop, both big and small things are tested. Let us say that a startup's early MVP is a website that has a short description of the forthcoming product and a button with text: "Click here to join our e-mail list if you got interested." The team makes different versions of the site. Some differences are small, such as the size of the mailing list button, while some differences are significant, such as the overall description of the product's key attributes.

3.2 Search for Business Model

So far this chapter has mainly discussed product development and how Lean can improve it. But simultaneously, as the product development goes on, a Lean software startup performs a crucial process: Customer Development. Creating a product without properly knowing its markets leads to financial disaster. Customer Development is the structured process of testing the entrepreneurs' guesses concerning customers, markets, channels, pricing etc. Turning assumptions into facts iteratively results in a decent business model. A business model is a detailed description of an organization's ways of creating, capturing and delivering value. [23]

The purpose of Customer Development is to make sure that there are going to be enough paying customers once the product is released, and to make the startup's customer knowledge keep up with product development. In the process, product development team will be interacting with the Customer Development team which really understands customer needs.

The Customer Development process consists of four cycles, each of which is repeated in an iterative manner before moving on to the next phase. The four steps are:

• **Customer Discovery** starts with the entrepreneurs' vision, which can be turned into a smart number of different business model hypotheses. Before doing anything else, there must be a plan of how to test these hypotheses and acquire facts. What matters at this point is the product specification rather than the features. The goal is to reach a situation where the developers and customers agree that there is a problem that needs to be

solved, and the specified product is going to solve this problem. Customer Development team members should go out of the building and see real customer reactions.

• **Customer Validation** tests whether the entrepreneurs' plan of doing business is repeatable and scalable. If their idea fails the test, it needs a pivot and the process steps back to Customer Discovery. During Customer Validation, product developers meet customers in the role of the pre-sales team.

• **Customer Creation** marks the launch of execution and is built on the company's initial sales success. In this phase the company scales its operations. Successful scaling is achieved by building end-user demand and navigate it to the sales channel.

• **Company Building** proceeds the execution and turns the startup into a company that follows its validated model. The product development team focuses increasingly on installations, as well as training the support and service staff.

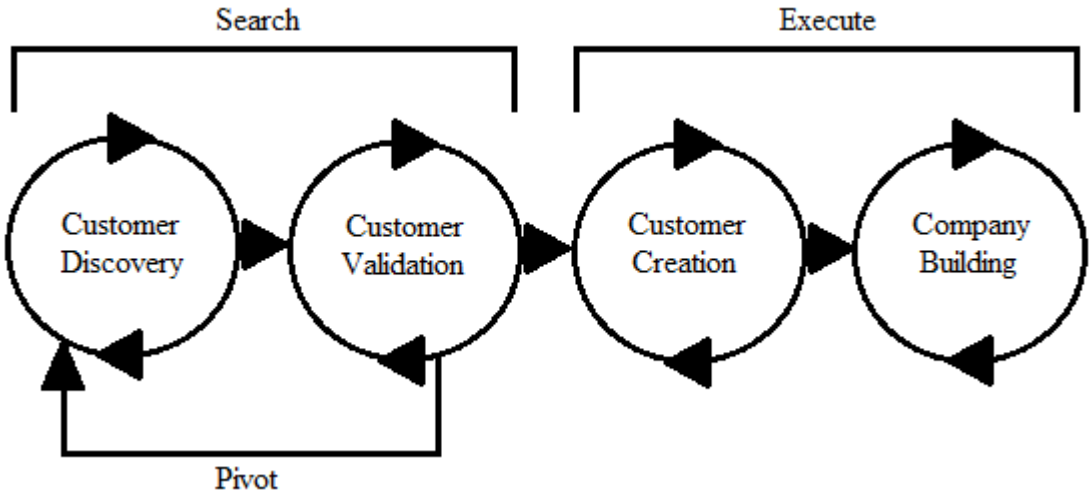

Figure 5. Customer Development process.

The four phases are illustrated in Figure 5. In every part of the process, the product development and Customer Development teams meet to discuss the situation and synchronize their work. Customer Development does not move forward to the next step until the teams agree. The first two cycles focus on searching the business model, while

the latter two are about executing the model. This thesis concentrates on the early stages of Lean software startups; the times of Customer Discovery and Customer Validation, and thus the creating of the business model.

The cyclic and iterative nature of Customer Development process is the main difference between this model and the traditional straightforward product development model. In the old days it was normal for entrepreneurs to strictly follow plans they had made, and going backwards was considered a failure. This method assumes that the business environment is both stable and predictable. The Customer Development model acknowledges that the world is neither of these. That is why the Customer Development process includes going backwards as a natural part of learning. It is acceptable to fail if you are going to learn from it. Having multiple iterations means more possibilities to learn.

Before starting the iterations, it is necessary to get familiar with one more factor that greatly appoints the very nature of a business model: the market type.

3.2.1 Startups' Market Types

Before any sales can begin, a startup must ask itself, "What kind of a startup is this?" In fact, be it a fresh new startup or a big established corporation, market type influences everything a company does. This includes product adoption rates, launch strategies, brand building and also how you evaluate customer needs. Market type is one of the factors that make startups differ from each other. Action tactics that work for one startup might be useless for the next-door entrepreneur.

Market type also affects the speed with which a startup moves in the Customer Development process. Completing the whole process may take more than one year. When a startup is going through the process, the importance of market type grows as the process goes further. During the first part, Customer Discovery, all startups should get out of the building and talk directly to customers. During Customer Validation, the differences between startup types emerge because sales and strategies show diversion.

During the third step, Customer Creation, the difference between startups' market types is even more critical as customer acquisition and sales strategy differ significantly between the market types. In Customer Creation, at the latest, a startup will fail if it has not understood its own market type. [24]

Market types are basically relations between product and market. Each of them has its own attributes, and the different types are: [24]

• Startups that are entering an existing market

• Startups that are creating a new market

• Startups that want to resegment an existing market with low cost

• Startups that want to resegment an existing market as a niche player

• Startups that clone a business model which has been successful abroad

New product in an existing market

Emerging in an existing market means that the product is doing something similar to what some previously released products are also doing. The competitors define the market, but thanks to them a new startup immediately knows who the customers will be. A startup falls to this market category if it rivals the competitors with performance and features.

One could also enter an existing market with a cheaper or niche-type product, but such markets are called resegmented market types.

New product in a new market

A startup creates a new market if it finds a large group of customers who had not been able to do something valuable before due to lack of suitable tools. A product also enters a new market if it solves a significant availability or convenience issue in a new way. For example the video website Vimeo created a new market in October 2007 by being the first service that enabled consumers to share and embed high-definition videos online. [25] This attracted serious filmmakers and professional musicians to use the

service. However, Vimeo was soon challenged by YouTube who added HD support in late 2008. [26] Today YouTube hosts more HD videos than Vimeo, much thanks to YouTube's community-centric social features. While there are no competitors for a startup that creates a new market, this advantage might be short-term. Furthermore, the users are unknown and unconvinced in the beginning. Successful creation of a new market requires a very well thought financing plan, because convincing the customers to adopt the new kind of product could take much longer than in an existing market.

New product attempting to resegment an existing market with low cost

Resegmenting an existing market may happen in two different ways: with a low-cost strategy or a niche player strategy. Low-cost resegmentation is exactly what it sounds like: the startup looks for customers who are willing to settle to a "good enough" product if it is cheaper than the competitors. Outside the IT business, the success of the furniture chain IKEA is an example of this strategy. IKEA products are not famous for being long-lasting, but consumers buy those anyway due to the affordable prices.

New product attempting to resegment an existing market as a niche player

Resegmenting a market with low cost is a rather straightforward strategy where it is clear for the entrepreneurs what to aim at. Resegmenting an existing market as a niche player is less obvious. Niche resegmentation looks at the market and asks, "Would some part of this market buy a new product that is designed to meet their specific needs? Maybe even if it cost more?" The revenue of niche players means a smaller share for the rivals, but niche players do not outcast market leaders. Niche startups survive with well-tailored and focused services that are desired by *some* customers in the market. Niche products do not even try to beat bigger companies in repertory.

A schoolbook example of a successful niche player is the American fast food chain In-N-Out Burger. Established in 1948, In-N-Out Burger entered a market that already had the bigger player McDonald's, which was founded eight years earlier. Unlike McDonald's who has a menu of dozens of choices, In-N-Out offered less than five items. The trick is that their products were made of fresh ingredients and were considered tasty. The company grew gradually and although In-N-Out Burger remains small in comparison with McDonald's, they have business in more than 250 locations today. [27]

Resegmenting a market as a niche player is arguably the best entry strategy for a startup. If a business is dominated by few big companies whose customers do not have much in common, chances are that they have left some specific product attribute with little attention. If a startup finds users who appreciate this particular feature, they might acquire loyal customers who bring enough revenue to get the startup growing.

In overall, resegmenting a market is more common for startups than creating a new market. [24] But no strategy is easy. For the creators of a low-cost product the goal is clear, but making a long-term plan about how to achieve low cost and remain profitable is tricky. For niche players, the resegmentation strategy has to deal with established competitors who wish to maintain their market shares.

Startups that clone a business model which has been successful abroad

Cloning a foreign business model means implementing a model that has been successful abroad, but has not been carried out in the entrepreneur's home country. In the world today, information spreads faster and more widely than ever before, whereupon getting to know foreign business models is easier to happen.

This market type is occasionally called "Copy to China", due to some well-known Western innovations that were copied in China. Among web startups these include such services as Renren, Tudou and Baidu Space, which resemble Facebook, YouTube and Myspace, respectively. However, the name "Copy to China" is misleading in the sense that there is no reason why an American or European startup could not copy an Oriental business model.

3.2.2 Customer Discovery

Too many startups blindly focus on their first customership, and only after the product is out of the building do they see that customers are not behaving as expected.

Customer Discovery consists of discovering whether the startup's hypotheses of the product, the problem and the customer segment are correct. At this stage the startup has to learn if the planned product is valuable to anyone, and if it is, who are the customers

and users. The purpose of Customer Discovery is to find problem/solution fit, which means discovering a valid and valuable solution to an existing problem. Whoever the customers are, they must find this problem worth solving.

A user or a customer will not be found inside the entrepreneur's office. You need to get outside the building already in this early stage of startup – this will also help you advertise your product to potential customers. This means the vision and the broad concept of the product; talking about individual software features at this point when the features are a subject to change would be both useless and questionable.

Because the product's upcoming feature set remains unknown during Customer Discovery, a Lean software startup does not write down a Marketing Requirements Document (MRD) for developers. In traditional marketing of established companies, creating MRDs is an important action, for those documents contain the sum of all customer feature requests and the documents are prioritized collaboratively by the marketing, sales and engineering teams.

In clear contrast to the MRD approach, a successful Lean startup's first release is designed to be just good enough for the *first* paying customers. No startup has the time and money to implement a product with every feature that an average customer wants in the first release. During Customer Discovery a startup should find visionary customers who are willing to be early adopters for the product. These customers understand that they have a problem which needs to be solved, even if it costs money, and they are actively looking for a solution to the problem. Most importantly for the entrepreneur, these customers see the potential in the startup's vision and they are willing to take the financial risk and purchase the product. These customers are called *earlyvangelists*. Quite possibly they have already had unsuccessful attempts to solve their problems with company's internal tools.

Once a startup finds earlyvangelists, the first release is tailored according to their wishes. The chronological order of these two actions is important: *first* you find customers for the product that you are already planning, and *then* you tailor some details of the release within the limits of the original product vision. In other words, the meaning of meeting customers so early is not to create a vision of the forthcoming product – the idea exists from the beginning.

What if no customers are found? In this case – and only in this case – the entrepreneurs hand a feedback list of requested features to the product development team. The features are included in order to finally find earlyvangelists, but although feedback from potential paying customers is valuable, their feature desires are considered as wishes, not as strict rules. This way the startup should avoid delays with the release.

The Customer Development process is divided into four phases that form an iteration. The same can be done to its first sub-process Customer Discovery. The phases are 1) stating all hypotheses, 2) testing problem hypotheses, 3) testing product concept and 4) verifying the results, respectively. The phasing is illustrated in Figure 6. The structure resembles the Learning Loop, for both consist of building (code / a list of hypotheses), measuring (customer interest / quality of the product concept) and finally learning. In this sense the Customer Discovery phase cycle, as well as any similar cycle, can be seen as an instance of the Learning Loop.

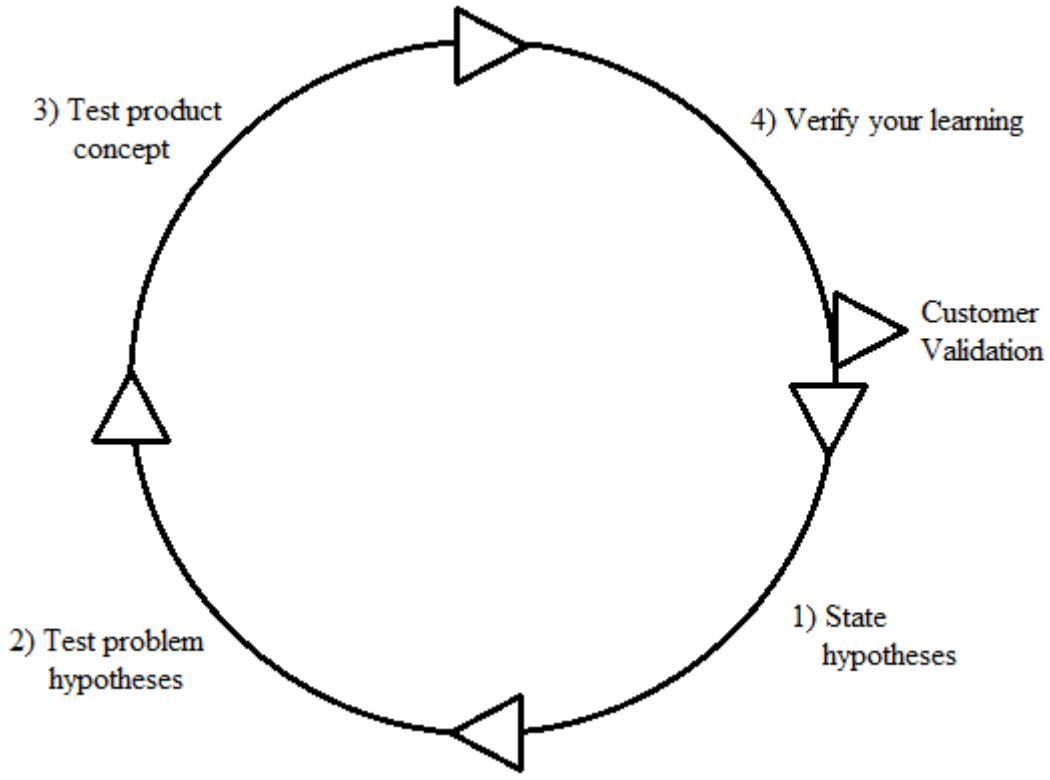

Figure 6. Phases of Customer Discovery.

The first step of Customer Discovery is to write down all hypotheses that the startup holds in its vision concerning customers and pricing.

The next job is to test the hypotheses in cooperation with potential customers. This is where you "get out of the building". The aim is to learn whether or not the startup has realized a real problem that is meaningful for the "ideal" customer. At this point the startup is not just randomly contacting any big potential customer, but is more interested in a limited number of customers who seem to fit the startup's vision of exactly right customer.

In order to truly understand the customer's problem, startup needs broad information concerning the customer's organization and business model. Still in the 21st century, physical face-to-face meetings are a good way to share information with each other. The startup entrepreneurs learn simultaneously about their customers and about what their own product should be like. The product vision is then updated with the learned knowledge.

In the third phase a startup tests its newly updated product concept in front of the customers with an MVP. This already might convince some enthusiastic earlyvangelists to say they want to buy the product, but the actual goal of this phase is to make sure that the product concept can now solve what it is supposed to solve.

In the last phase the startup checks once more that it now understands what problems the customers have, and that the customers are ready to pay for the solution, and that the planned pricing will cover all expenses and make profit. Otherwise Customer Discovery needs another iteration.

When Customer Discovery is considered done, it is time to write the deliverables for the next Customer Development sub-process, Customer Validation. These deliverables are a problem statement document, a product requirement document, a revenue plan, and solid business and product plans.

As a short summary, an iterative cycle of Customer Discovery gives answers to these questions: [28]

• Have we identified a problem that a customer wants to get solved?

• Is our product a solution to these customer needs?

• Have we learned enough to go and sell now?

If entrepreneurs find all three answers to be “yes”, it is time to move on to Customer Validation. Otherwise there is a need for further iterations of Customer Discovery. [28]

3.2.3 Customer Validation

In the Customer Development process there is a feedback loop from Customer Validation to Customer Discovery. This pivot means that in the Customer Development process, Customer Validation is the final checkpoint for verifying that customer needs (problem and its solution) are understood, and that the startup has a realistic and profitable plan of how to sell the product. If a startup fails in finding paying customers during Customer Validation, the process returns to Customer Discovery in order to (re)discover what customers are ready to pay for.

It is generally assumed that most startups that follow the Customer Development model will perform at least two iterations of both Customer Discovery and Customer Validation. [24] This may seem expensive, but in the long run it pays off, for the first two sub-processes of Customer Development process burn less cash than the remaining two, namely Customer Creation and Company Building. If the business model proves to be incorrect, it needs to be fixed sooner or later anyway, and repairing a small startup is cheaper than reforming an accelerating company.

The goal of Customer Validation is to find product/market fit: to be in a good market with a product which satisfies that market. [29] When this fit is achieved, the startup has not only built a product that many people want, but it has also validated its business model. Any startup must be aware of whether it has found this asset or not. Luckily, measuring that is rather easy. If the customers do not get value out of the product and closing deals is overly difficult, the startup has not found the product/market fit. Consistently, if the product is gaining popularity as expected and the startup holds a repeatable and scalable business model, the product/market fit has been found.

At this stage the startup creates a repeatable roadmap for the sales and marketing teams. This sales roadmap includes authentic, successful and repeatable sales cases where the customers buy the product. This is when the startup should find some earlyvangelists. If they cannot be found at this point, the situation will not get any better in the future. In fact, the startup should find enough earlyvangelists to recognize a set of customers, as well as a profitable market.

Since Customer Validation (as well as Customer Discovery) takes place before the startup grows big, it is only logical that the founders should lead the Customer Validation team. Like said before, Customer Validation is a process where the whole business plan can still undergo remarkable changes. It is important that the entrepreneurs get feedback directly from the customers. Although the sales roadmap will be utilized by sales and marketing teams, creating the roadmap must not be delegated to people who only work in these teams. [28]

Creating the sales roadmap uses all of startup's validated learning to answer a number of questions. Compared to the question list in the end of Section 3.2.2, the focus is now switched from customer's problem to the decision of buying or not buying, for the questions for sales roadmap are:

• How long does a normal sale take from beginning to end?

• Who is the decision-maker?

• What is the selling strategy? Is this a solution sale?

• If it is, what are the key customer problems?

• What is the profile of optimal earlyvangelists?

Much like Customer Discovery, Customer Validation can also be shown as a cycle of four phases: 1) get ready to sell, 2) get out of the building, 3) develop positioning and 4) verify your learning. See Figure 7, Phases of Customer Validation.

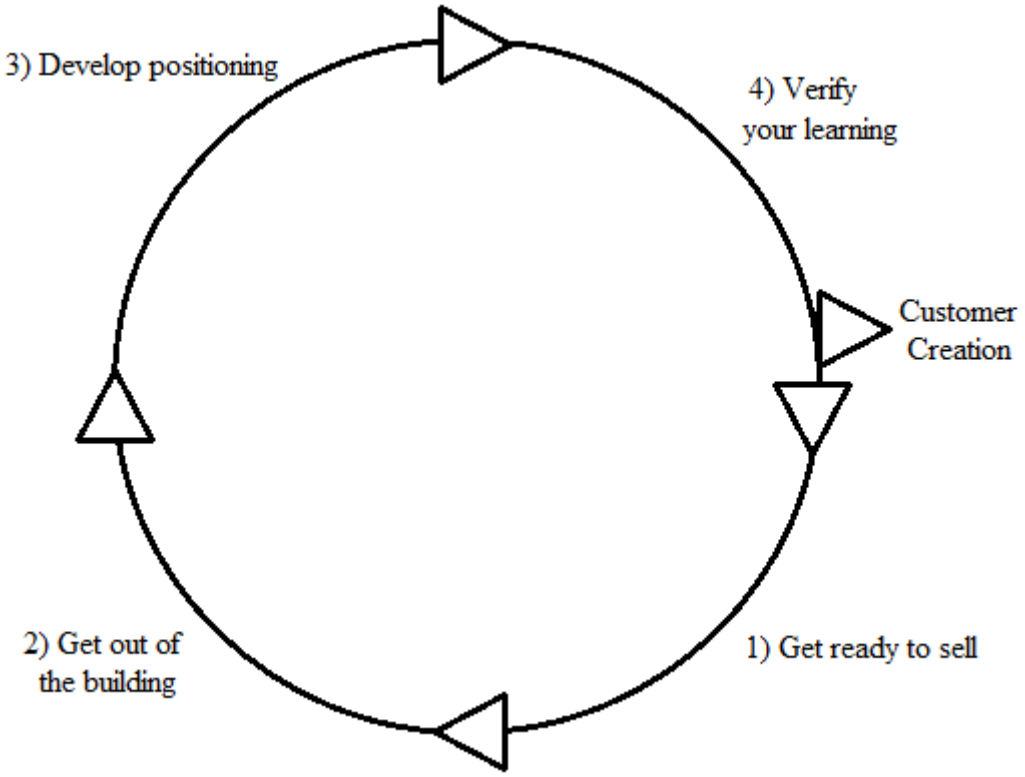

Figure 7. Phases of Customer Validation.

"Getting ready to sell" means some very practical activities, most importantly product positioning, creating a high-fidelity MVP, gathering plans to activate customers and building the metrics toolset.

Product positioning is not successful until the startup is able to explain shortly for whom the product is targeted, what is the reason to buy it, and how it is different from possible competitors. This statement is a subject to change in the later phases, as the startup gets feedback from customers, analysts and investors.

The startup need a sterling MVP for the customers, but the MVP does not appear from a scratch at this point. It was already built in phase three of Customer Discovery. In Customer Validation it is just polished into a high-fidelity form.

During Customer Discovery the startup made prefatory plans to acquire and activate customers. Those plans are now improved and activation tools are built. In practice, acquisition and activation plans are nothing but guesses and tests, such as A/B-version testing. Therefore the activation tactics should change at a brisk rate.

The importance of testing all assumptions outside the startup building (or room or corner, for that matter) has been repeatedly mentioned in this work. In the second phase of Customer Validation, entrepreneurs get out once again to see whether their plans to activate customers work or not. For this experiment the startup need its high-fidelity MVP and activation plans from the previous phase. If only few potential customers are willing to activate and pay, the business model is not repeatable. A startup also needs the unwilling customers to tell them why they do not find the product attractive. Getting paying customers and getting feedback are equally important.

The third phase, "develop positioning", begins when the second phase is carried out well enough to get the startup at least a couple of orders. In this context "positioning" refers to both product positioning and company positioning.

The startup already has a short product positioning statement which has been refined in both Customer Discovery and earlier phases of Customer Validation. By now the startup has a fair amount of customer feedback and it is time to check if the positioning statement matches real-life customer reactions. If the product positioning statement is not compatible with reality and the entrepreneurs do not know how to refine it, they contact the customers and ask them directly what is wrong.

The other half of the third phase is company positioning. The difference to product positioning is that the latter focuses on product attributes, while company positioning answers company-related questions: "What good does this startup do for a customer?", "Why would a customer want to do business with the startup?" and "Why does this startup exist and how is it different from all the others?". Obviously these questions, especially the first one, connect company positioning to product positioning.

Just like the product positioning statement, the company positioning should be available in a brief and validatable form. If the startup entrepreneurs naively believe that customers want to do business with them because their startup is the best of all, their company positioning statement cannot be validated – "best" is too abstract a word. Instead of that, attributes like speed, price and reliability are comparable and those can be used when talking about company positioning.

While refining the product and company positioning statements, entrepreneurs must always keep their startup's market type in mind. Positioning is completely different

depending on whether the startup is entering an existing market or creating a new market. Positioning statements should be so clear that the market type could be deduced correctly just by reading them.

In the fourth phase of the Customer Validation cycle, the entrepreneurs sit down to think thoroughly if the startup should pivot or proceed. At this point the startup is still a small company, meaning that potential losses are relatively small and changing the business plan is simple. Taking some time now to consider the pivot-or-proceed question pays off in the long run. The entrepreneurs must ponder is their business able to generate revenue and have they learned so much that they can scale the operations. [28] If they can, the startup moves on from the "Search" to the "Execute" part of the Customer Development process (see Figure 5). The scope of this thesis is in the early days of software startups when the business model is being searched, whereupon this work crops out the contents of the "Execute" phase.

3.2.4 Business Model Canvas

The previous pages suggested that startup founders should ask themselves several questions, such as "What good does this startup do for a customer?", "Why would a customer want to do business with the startup?" and "Why does this startup exist and how is it different from all the others?" To answer these kinds of questions, entrepreneurs need a clear strategic way to perceive the key elements of their own business, and a way to compare their business model to the rivals'.

One significantly popular tool for describing and developing the business model is the Business Model Canvas (BMC). It was created by Alexander Osterwalder in 2008. [30]

The canvas is useful for both building and reforming the business model. It can help entrepreneurs to realize how they plan to make money and what they should change. The canvas is a fast and compact tool: writing a business plan document may take days or weeks, but with a one-page template entrepreneurs can outline several business models in a few hours. The one-page format also forces to get directly to the point.

The canvas, shown in Figure 8, suggests that a business model consist of nine main blocks that together describe the company's strategy: Customer Segments, Value Propositions, Channels, Customer Relationships, Revenue Streams, Key Resources, Key Activities, Key Partnerships and Cost Structure. [31]

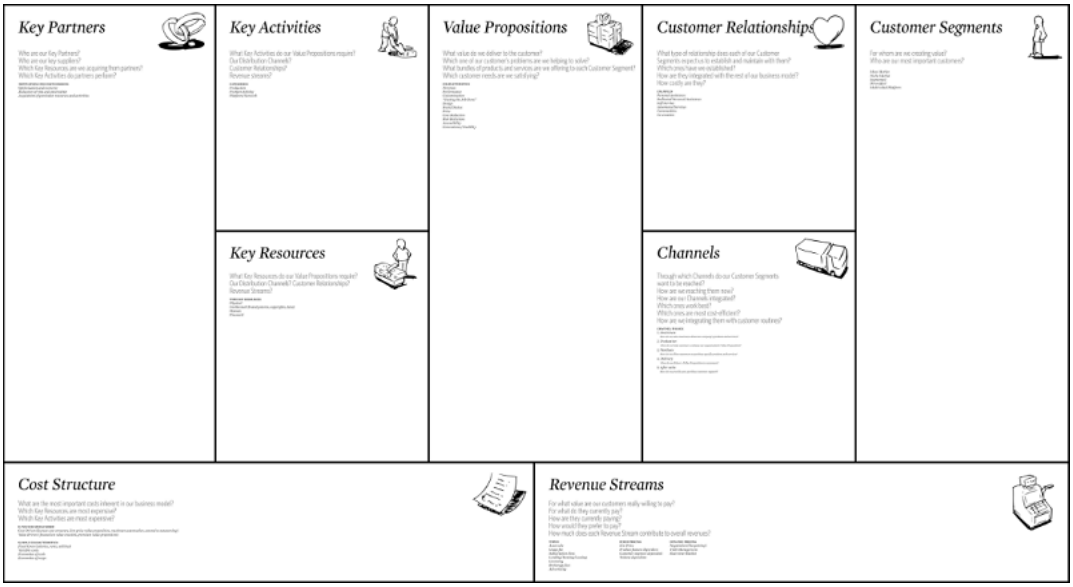

Figure 8. Business Model Canvas. [31]

Osterwalder himself has compared the Business Model Canvas to a theatre. When you look at Figure 8, you can think about the blocks on the right side as a front stage. Areas like Customer Relationships and Customer Segments represent interaction with the customers. On the other hand, the left side of the canvas is like a theatre's back stage. Customers rarely see a startup's Key Partners and Key Resources, but all that is needed for the show to go on. [32]

A company serves one or more Customer Segments. If some customers are willing to pay for different services/products, they have different needs and therefore belong to separate Customer Segments. Already on its day one a startup has a vision about which segment(s) to serve. Like every other block in the canvas, this may be reformed later on.

Value Propositions satisfy the needs of Customer Segments. Value Propositions are the reasons why customers want to do business with a company. They may be qualitative

such as user experience, or quantitative such as speed. They are either new innovations or more attractive versions of existing services.

Value Propositions are delivered to customers through Channels. The Channels block of Business Model Canvas explains how the company communicates with the customers and provides Customer Segments with Value Propositions. One could call Channels the interfaces between customers and the company. A company can build its own direct Channels and/or use already existing indirect partner Channels. As always, the optimal solution depends on the type of the company. If a company needs to reach as many customers as possible, it is reasonable to utilize all available partner channels. If the company aims at high margins from every sale, it is good to minimize the number of intermediates.

While Channels are the passages of communication and delivery, Customer Relationships are the types of relations between the company and the Customer Segments. There is a wide range of possible relationships, all the way from face-to-face meetings to automated data transfer.

Being able to deliver value and making deals with customers result in revenue. The Revenue Streams block tells what are the Customer Segments paying for and how much does each stream bring revenue. Pricing mechanisms may differ in each Customer Segment, for there are several methods to create Revenue Streams, e.g. usage fees, subscription fees, licensing, renting and advertisements.

Key Resources are the most essential and necessary resources, without which the business model will not work. They differ from company to company, but these resources are something that enable the company to create value to the customers and maintain good relations with them. For a software company Key Resources are commonly intellectual resources (copyrights, patents), human resources (high-level education) and financial resources.

Companies need Key Resources to successfully carry out their Key Activities, the most important actions in the business model. Customer Segments pay for the direct or indirect outcomes of the Key Activities, meaning that a company will not have profitable Revenue Streams if it fails with its Key Activities. For software companies

these activities include processes like problem solving (software design), production (coding and testing) and platform-specific actions. [28]

The second last block is Key Partnerships. For a software startup that has not outsourced any activities nor has any suppliers, it is possible to keep this block rather blank.

The last block of Business Model Canvas is Cost Structure. If Business Model Canvas is again imagined as a theatre, Cost Structure summarizes how much it takes to run the backstage, in other words the operational part of the company. This includes the costs of acquiring Key Resources, creating value and maintaining relationships. Some costs remain constantly the same, while others vary. The business model is profitable if and only if total revenue is greater than all costs combined.

3.2.5 Lean Canvas

The nine blocks of the Business Model Canvas cover a wide range of strategic elements and the Canvas is informative not only to entrepreneurs but also to customers and investors. However, entrepreneurs may understand their own startup better if they use a tool that is targeted just for themselves. The Business Model Canvas shows a startup's strategy, but the tactics beneath that strategy are not presented in a very actionable way.

Osterwalder created the Business Model Canvas in 2008. One year later Ash Maurya published the Lean Canvas (LC), which was inspired by the works of Osterwalder and Rob Fitzpatrick. [33] Maurya's goal was to create a business model documentation template that enables an abundant amount of learning about the practical things that startups need to consider in their early days. The Lean Canvas is shown in Figure 9. It is adapted from the Business Model Canvas, so there are both differences and similarities. The canvasses share five similar blocks: (Unique) Value Proposition, Channels, Customer Segments, Cost Structure and Revenue Streams. The right and left sides of the Business Model Canvas were already compared to a theatre's front stage and back stage. In the Lean Canvas this structure is explicit: the right side focuses on market and the left side on product.

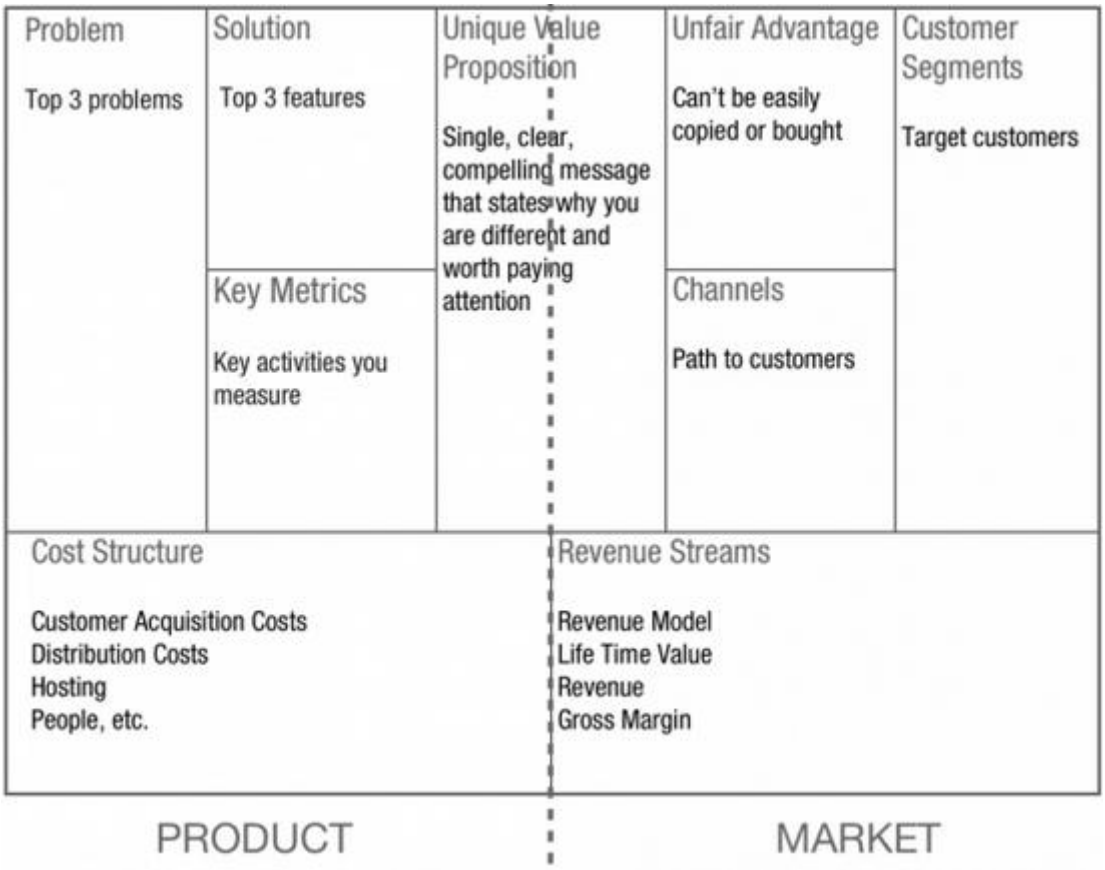

Figure 9. Lean Canvas. [34]

The first difference between the two canvasses is that the Lean Canvas includes Problem as an independent block instead of just implying customer problems with Value Propositions. A problem cannot be solved until it is properly identified.

The second Lean Canvas block, Solution, includes one solution for each top problem. Early understanding of customer needs helps to build the right features and save time. The Lean Canvas encourages to list only one to three most important problems and solutions, which is compatible with the MVP concept. Solutions are confirmed with early customer feedback. Solution "replaces" the Key Activities block of Business Model Canvas, because Key Activities can be derived from solutions.

The third block, Key Metrics, should also be kept small. There is a myriad of possible variables to look at, but only few of them matter for a startup at any given point of time. The outcome of failing to recognize the key metrics is wasting resources. Key metrics differ from one startup to another, depending on market type and engine of growth, and

they change with time. Every hypothesis that is tested requires a metric to measure it, but not everything should be tested simultaneously. The Key Metrics box is intended to state the metric(s) that are most relevant in the current situation. Maurya suggests that Key Metrics are more informative to entrepreneurs than Key Resources, while the latter are relevant to outsiders who try to understand what the startup is doing. [34]

The Unfair Advantage block consists of any competitive advantages that cannot be easily copied or bought. Most early startups have none. The block is mainly meant to encourage and remind entrepreneurs to build some kind of Unfair Advantage over time. If a company never finds an advantage, its customers may be taken by competitors. For example Facebook's social features are easy to copy, but the network's huge popularity is an Unfair Advantage that a new startup cannot possess. Other than this block, the Lean Canvas pretty much ignores competitors. The canvas is a tool for entrepreneurs to define their startup, and those who decide competitors are not the entrepreneurs but their customers. In fact, the best way for *any* company to identify competitors is to look at what the (potential) customers are doing.

Compared to the Business Model Canvas, Key Activities, Key Resources, Key Partners and Customer Relationships have been removed/replaced, and Value Propositions are now called Unique Value Propositions. The cases of Key Activities and Key Resources were already explained. Key Partners are removed because if a new startup puts effort into building partnerships before even testing the product, the effort might be wasted. Later on Key Partners can be highly important, but the Lean Canvas puts focus on the early days.

The absence of Customer Relationships in the canvas is explained with overlap. Best forms of Customer Relationships depend on Customer Segments and Solutions, and relationships are acknowledged as paths to customers. Therefore they are captured by the existing Channels block that covers both free and paid channels. [34]

Unique Value Propositions (UVPs) are compact messages that state why this particular startup is worth the attention. A good UVP tells what the product is, who the target customers are and why they should pay for the product. The last marketing statement may sound similar to the Problem block, but the difference is that Problem captures the

customers' top problem(s) while UVPs are marketing promises, such as "we'll fix your broken computer in seven days or the repair is free of charge". [29]

The very first things that matter to a startup are identifying a problem worth solving, finding a valid solution and realizing who the customers will be. Then the startup needs to realize how it is going to differ from other companies. There is nothing wrong about drafting all blocks on startup's day one, but when the startup is trying to reach the problem/solution fit, it should primarily focus on Problem, Solution, Customer Segments and Unique Value Proposition. Thinking about best channels and other things can and should wait.

Which one is found first: the problem, the solution or the customer segment? This differs from one startup to another.

One possibility is that the software professionals first identify a problem and then found a startup which aims at building a product that solves the problem. In this scenario, paying customers are found last – if they are found at all. Lean startup methodology makes this risk easier to take, because it is meant for increasing productivity, learning fast and minimizing the time spent. If the startup fails, perhaps it fails so fast that the entrepreneurs have not left their other jobs.

Another approach is to find the customers and their existing problem simultaneously, and then create the solution. This requires good connections and a considerable amount of customer-related information from the very beginning.

4 Metrics and Tests

In the search for a business model, testing and optimizing are always immanent. This practical chapter takes a look at what exactly should be tested and optimized. Any startup will run out of resources if it tries to observe and test everything without prioritization. Often there is a need for a large number of different kinds of tests, so it is normal to switch from one test to another rapidly, but testing should be planned and controlled. Planning reasonable tests requires understanding which metrics are the most important.

4.1 Vanity Metrics and Actionable Metrics

Having lots of tests means getting lots of data, but having more data is a good thing only if the startup knows how to look at it. Otherwise, executing tests is practically waste. A startup must realize what information is relevant for success and what is not. If there is a piece of data that will not affect any decision-making, it is worthless. If a piece of data tricks entrepreneurs to believe that their startup is doing better than it actually is, that data is below worthless.

Some metrics deserve attention while others do not. The main question is: “Does this metric help the startup to make reasonable decisions?” If you do not know what to do when you see a metric, it is most likely a vanity metric.

Vanity metrics are data points that do not contribute decision-making. Since people see what they want to see, vanity metrics often give an overly rosy image of reality. For example, a software startup might have a promotional website for its product. The startup changes many things on the website in one go: layout, colors, text content and images. A few days later the number of unique visitors doubles. In this situation the number of visitors is useless information for the startup, because entrepreneurs do not know what caused the increase. Perhaps changing the text content made the page more attractive for online search engines. Perhaps the new layout made the site look really

interesting and visitors began to link the site to their friends. Or perhaps some popular blogger happened to mention the product by name on his blog.

Another example: Lean manufacturing faced vanity metrics issues when it was first introduced in factories. Back in those days, factories were expected to keep the machines working at full capacity as much of the time as possible. From the perspective of an individual machine this is efficient, but from the viewpoint of the entire factory it does not guarantee great productivity. If one machinery process is slower than all others, and if each product has to go through that process, keeping all other machines running all the time is waste. [18]

Startups are expected to gain profit – not on the day of founding, but sooner or later. More than anything else, a startup needs to observe where the revenue is coming from, and what is consuming the revenue. In the scope of this thesis, this boils down to a few questions:

• What are the key benefits that customers are coming to us for?

• Why do we gain or lose customers?

• Why do we gain or lose revenue?

• What costs do we have?

Metrics that answer these questions are called actionable metrics, for they trigger actions (i.e. decisions). Important actionable metrics include at least the number of weekly/monthly transactions, average order value, net profit, lifetime value (the total amount of money the startup ever gets from an average customer), customer acquisition & activation cost (gets easily high if the startup utilizes paid advertisement), market type, burn rate (how much money is the company using in a month and how long will the money last) and of course the total monthly revenue. [35] Some actionable metrics, for example cost structure, were mentioned as blocks of Business Model Canvas in Section 3.2.4. There are others too, but the relevance depends on the company and its engine(s) of growth. For a startup whose engine of growth is sticky, it is highly important to keep close eye on customer satisfaction surveys to keep the customers. Or if a startup is planning to have a viral engine of growth, it must monitor how many new

customers each existing customer brings to the company. The growth is called viral only if the average customer brings at least one new activated customer.

Figure 10 shows the metrics of a fictive anonymous Lean software startup. This example is very much simplified but serves the educational purpose. The startup is asking for mailing list subscriptions and product pre-orders on its website. The black line (number 1 in Figure 10) indicates the number of daily visitors on the website. The dark grey line (2) is the daily number of new e-mail subscriptions, and the light grey line (3) is the daily number of pre-orders. The scale of the vertical axis is irrelevant.

In the beginning (day A) almost all page visitors are ready to subscribe to the startup's mailing list, but few of them want to pre-order the product. On day B the startup pays for large advertisements in the local newspaper for four days in the row. This immediately makes the number of visitors jump, but that is a vanity metric in this case: the number of new pre-orders increases so insignificantly that it does not cover the advertising budget.

On day C the newspaper advertising has been going on for two days and all readers have already seen the startup's ad. They start to lose interest and all indicators start going down. On day D the advertising is over and everything is just like in the beginning. If the startup focused on line 3, it has now learned that newspaper advertising is not worth it. If the entrepreneurs only looked at line 1, they have not learned anything.

On day E the startup's webmaster adds one line of text on the website: "We will never give your e-mail address to any third party, nor send you spam." This encourages more people to give their contact information to the startup. Soon afterwards on day F the startup sees an increase in the number of visitors. This is because while the page gets new visitors just like before, also some of the old visitors return to the page as they are reminded via e-mail. (What makes this example unrealistic is that the previous number of mailing subscriptions didn't have *any* impact on page views; the reason was to keep the example as simple as possible.)

Now the number of visitors and daily e-mail subscriptions are in a slow but steady rise, but the actionable metric, which is product pre-orders, has not improved at all. On day G the website's "Order by clicking here!" button is made bigger and more colorful. This

does not help, but the entrepreneurs learn that they must try something else. Then on day H they modify the product description text into a more clear and more compact form, so that every visitor has the time to read it through and the capability to understand what the product is made for. Now the number of daily pre-orders finally starts rising. What made the learning possible is that the startup didn't change many things at once, so when something happened the cause was always obvious.

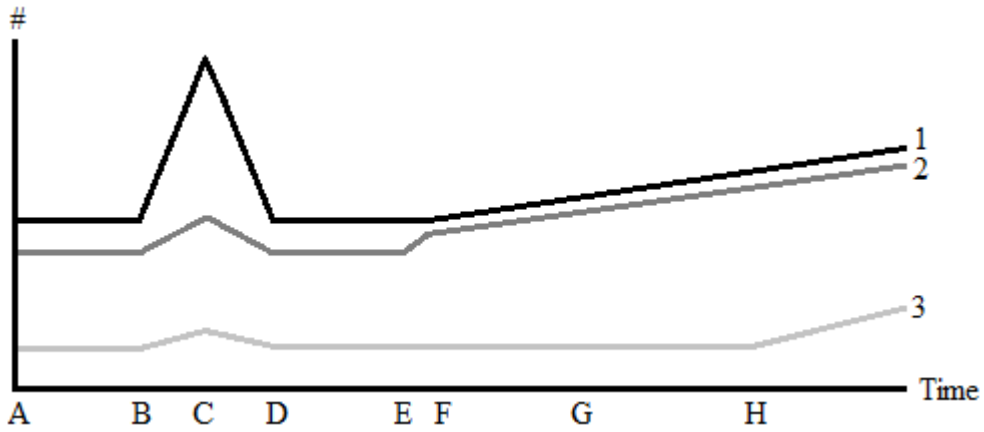

Figure 10. Metrics of a fictive Lean software startup.

4.2 Meaningful Testing

Early Lean startups which have just got started with the Learning Loop face major uncertainty. The upside of this is that when you do not know much, you need relatively little data to decrease uncertainty significantly.

At any fixed moment, a startup should focus on only one metric. Whatever that metric is, it must be clearly acknowledged. It is possible to handle multiple metrics and tests at the same time, but this can be complicated and misleading because recognizing causality becomes harder. For this same reason, whenever some test is being executed it is important to run it the same way until one or more repeating patterns come out.

What matters just as much as a metric, is the usage of it. Metrics do not explain themselves. Any metric becomes a vanity metric if it is used only to document the current situation and if it offers no insight into what to do next. Metrics can help

entrepreneurs realize what is going right or wrong, but they will not tell why. Startups need to contact users and customers to know what they could do better.

Tests are meaningful only if the hypotheses behind them are falsifiable. A falsifiable hypothesis is a specific statement that can be proven wrong. A statement like “investing a bit of money in advertising will get us a smart number of new users” is too abstract, because “a bit of money” and “a smart number” are unspecific terms, and the medium of advertising remains unknown. On the other hand, a hypothesis such as “our next blog entry will trigger 50 new signups tomorrow” is falsifiable because it is testable and specific.

The fastest way to test something is talking to customers. Direct interviews in a natural language offer more comprehensive feedback than collecting analytics. Interviews are good for recognizing the customer’s problems, finding solutions to those problems, and testing MVPs. A startup may not have resources for a myriad of customer meetings or even phone calls, but as few as 5–10 interviews can teach a lot. Mathematically such a number may not be significant, but if five out of five customers say that the new feature in the updated MVP is working horribly, that is quite significant. Dissatisfaction could also be recognized with a poll, but in interviews it is more natural to give reasons for the answers. [29]

Qualitative methods, e.g. interviews, are fast ways to learn, but startups need both qualitative and quantitative testing. The latter means signup statistics, order rate analytics, online polls etc. Relying only on qualitative data can cause false positive results, meaning that a hypothesis is seemingly but not actually validated. While unanimous negative feedback shows that the startup needs to fix something, clearly positive qualitative signals do not automatically indicate that the hypothesis is correct. Let us say that a startup has ten customers, out of whom one is currently satisfied while the remaining nine are not. Entrepreneurs interview one randomly chosen customer. There is a small yet real 10 % possibility that they happen to contact their only satisfied customer and do not get the valuable negative critique they need for learning.

Every startup is unique. They have different market types, hypotheses and engines of growth. There is no universal rule about what exactly should be tested by an early startup. However, for a software startup *potentially* meaningful aspects include at least

the following: 1) acquisition, 2) activation, 3) retention, 4) revenue and 5) referral. These metrics indicate 1) how many users find the product, 2) are they impressed by their first user experience, 3) do they use the product repeatedly, 4) where the money comes from and 5) do users tell other potential users about the product. These are called Pirate Metrics, because when you put the first letter of each metric together, they spell the word AARRR. [29]

Acquisition happens when someone gets interested in the product. This interest can be measured in a number of ways, such as the download counts or the number of mailing list subscribers. Even before dreaming of revenue, startups need enough acquisition to support their learning.

Activation is the point where an interested customer has his first positive user experience. This experience must be so satisfying that the user actually tests the product instead of quickly abandoning it.

Retention measures engagement with the product, for example opening software again or logging in on an online service.

Revenue covers all actions that bring money to the startup.

Referral measures how much the users introduce new potential users to the product. For startups with the viral engine of growth, this is definitely a meaningful metric. Compared to revenue, referral can be difficult to measure, for people may talk about the product without the startup knowing about it.

The Pirate Metrics form a funnel where you first get customers, then focus on keeping them and finally you try to "make them grow", meaning that with time you receive more revenue and new customers thanks to a viral loop: existing customers attract fresh acquisitions. This funnel and the loop are illustrated in Figure 11. At first the funnel gets narrower, as only some acquisitions lead to activation and only a fraction of activated customers become engaged with the product. On the other side the funnel gets larger again, meaning that those customers who were successfully engaged generate continuous revenue and make the startup's product better-known. The slopes of the funnel differ significantly between different startups. [36]

In the funnel, the conversion rate from one step to another gets smaller and smaller. It might well be that 100 people become interested in the product, 70 of them try it at least once, 30 people enjoy their first user experience, five people return, two of them pay for the usage regularly and only one makes a reference to the product while communicating with potential future users. However, a startup might survive even when the funnel's right side is sloping rather gently. Conversion rate goes down, but at the same time the average income per user goes up.

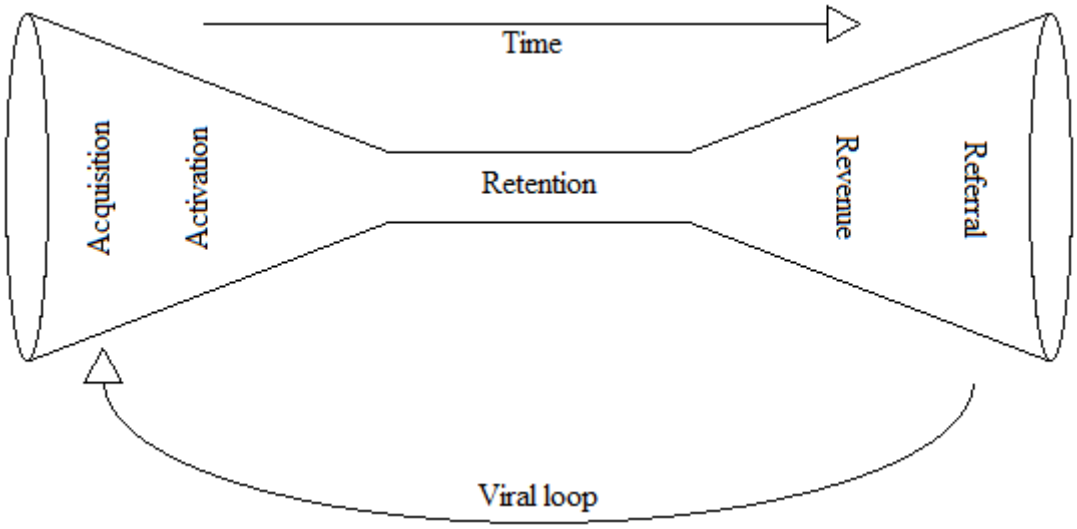

Figure 11. Funnel of Pirate Metrics. Based on the work of Ash Maurya. [29]

Statistics and figures about Pirate Metrics have been used in marketing for a long time, but Lean startups use them also in product development. [1] Knowing the percentage of acquisitions that lead to activation, the percentage of activated users who become long-time users, etc., helps the startup realize where the product is losing users most seriously at the moment. The focus of development then shifts to that point.

5 Case Studies

5.1 What was Researched

Four Finnish Lean software startup entrepreneurs were interviewed for this work in May and June 2013. The primary motivation for the interviews was to get qualitative information about how compatible the Lean startup methods are with the real world and how good or bad results the entrepreneurs have reached with them.

The interviews were divided into six parts. The first part discusses what the startup is doing and where it is right now. The second part reveals how the startup sees Lean startup methods and their value. Part 3 is about the startup's business model and how it is been created. Parts 4 and 5 strictly focus on this work's emphasis: going through Customer Discovery and Customer Validation. The sixth and final part is where the interviewee looks back to say whether Lean startup methods have been worth adopting. He also gets a chance to speak his mind freely. All these questions, which were presented to every interviewee, are listed in Section 5.2.

All interviewees were told that they can skip any questions that they do not want to answer. However, none of them used this opportunity. The interviewees did not affect each other in any way, for all interviews were done on different days and the interviewees' names remained confidential.

Section 5.3 introduces the examined startups and takes an insight and analysis into the given answers. The section provides real-life information about the practicality and usefulness of the ideas that this work has discussed in previous chapters. In Section 5.3, some questions get discussed more than the others, based on how valuable answers the interviewees were able to give.

Section 5.3 is divided into six parts, which mostly correspond with the grouping of the questions in Section 5.2. However, 5.3 should be read as a whole because the startups have been doing Customer Discovery and Customer Validation simultaneously.

5.2 Interview Questions

The following questions, grouped into six clusters, were asked from the four entrepreneurs who were interviewed for this work.

1) Background information

- Who are you? Which startup do you represent? What is your position in the startup?
- What kind of software does the startup do?
- How was the startup funded in the first place?
- How many staff members does the startup have?
- In what kind of a situation is the startup at the moment?

2) Perception of the Lean startup concept

- How would you describe the Lean startup concept with a few sentences? How is it different from other development philosophies?
- Why do you think that the Lean startup philosophy is worth using in software startups?
- How well-known is the meaning of the term *Lean startup* among Finnish software professionals at the moment?
- Where does this startup's Lean startup -related knowledge come from?
- Are there any potential risks or downsides in implementing Lean startup methods?
- How early did this startup start implementing the Lean startup thinking? Was it an easy decision to make? Why?
- How has this startup concretely utilized the Lean startup concept?

3) General questions about the business model

- What is the (potential) market type of this startup?
- Which engine(s) of growth does this startup have?
- Is this startup selling B2B or B2C?
- Which tools have you used to create the business model? How valuable do you find them?
- Have you implemented the Learning Loop?

4) Customer Discovery

- How clearly are Customer Discovery and Customer Validation separated from each other in your startup?
- Which hypotheses did this startup have in the beginning? How much was provably known on day one?
- How were the original hypotheses tested? Which tests and metrics were used? Were they meaningful? How did you validate your learning?
- What is your plan to acquire and activate customers? How do you keep them active?
- What was the startup's first MVP like? Was it overly developed; could you have released it earlier?
- How long did it take to find the problem/solution fit? Was there something you could have done more effectively in that?
- Which was recognized first: the problem to be solved, the solution or the customer segment?

5) Customer Validation

- Product positioning: Whom is the product made for? Why would someone buy it? How does the product differ from the rivals? Has your product positioning changed during the startup's lifespan?

- Company positioning: Why does this startup exist? Why would a potential customer choose this particular company? Has your company positioning changed during the startup's lifespan?
- How did you activate customers during Customer Validation? How did you test your activation methods? Did you have a high-fidelity MVP at this point?
- How actively did you gather customer feedback during Customer Validation? How important did you find the feedback? How was the feedback analyzed?
- Did you pivot? If you did, what triggered the decision to pivot? How did you pivot? How regularly did you think about pivoting?
- Is there something about Customer Validation that you could have done better?

6) Miscellaneous

- Do you think that implementing the Lean startup concept has paid off?
- Would you do it again?
- Is there something else you want to say? What?

5.3 Interview Results

The four contacted companies are referred to as A, B, C and D throughout this chapter. All individuals who represent these companies in the interviews are their founders, CEOs or main designers.

1) Background information

Startup A is creating a tool for marketing analytics. The service is targeted for companies that are doing online marketing and want to discover which segments of their business are underperforming. The first released version of startup A's product came out just one day before the founder/CEO/main designer was interviewed. While

he is the only full-time worker, there are also a few part-timers as developers and salespersons. So far A has been running on the founder's personal wealth, but the startup is currently looking for outside funding to hire more full-timers.

Startup B provides web companies with cloud-based services for testing user interfaces. The goal is to make the service highly automated and very easy to use. The startup was founded in late 2011 and consists of two founders, one full-time intern and two part-time programmers. One of the founders was interviewed.

Startup C is a spin-off of a software consulting and web hosting company that was founded already in 2004. Startup C was detached from that company during 2009. Startup C has created a communication tool for software developer teams. These teams can work in various companies. C was sold to a considerably big international company three months before the interview, but according to C's founder C is practically still operating just like before. By the time of the interview C had nine employees and two more were just about to step in.

Startup D designs retail analytics for improved performance marketing measurement. Their products are made for retail chains, airports, malls and any place where there is a need to monitor the behavior of anonymous customers. The startup has been doing this since 2010 and consisted of eight people in June 2013. D's plan is to double the number of staff members until the end of 2013. That intention is highly ambitious compared to the other three companies.

2) Perception of the Lean startup concept

All of the four interviewed entrepreneurs see the Lean startup concept in a very similar way and use just slightly different words and emphasis when asked for a short description of the concept as a whole. The Lean startup philosophy is understood as a way of optimizing the process of creating a healthy business model. What makes the concept particularly good for real-life cases is that uncertainty is accepted as a fact, and the importance of fast customer feedback is highlighted. Bad ideas fail before they consume all available money, because different solutions and their internal details are tested frequently. Startup D's CEO pointed out that the Lean startup methods are

particularly meaningful for Finnish entrepreneurs, for the Finnish business culture has a tradition of first building a product and only then getting out of the building to look for customers.

Any given software company, especially a startup with humble resources, should focus on work that is actually valuable. This is intuitively clear on the theoretical level, but in practice most companies perform some waste activities that do not add value to the product. The interviewed entrepreneurs have experienced several of these cases. They are glad about how the Lean startup framework gives systematical tools for validating what the customers are willing to pay for.

The interviewees explicitly claim that the meaning of the term *Lean startup* still remains unclear among Finnish software professionals, though the term has been widely used internationally for several years now. There seems to be a generation gap in Finland: young software workers have at least some information about the concept, while the older ones have at most heard the term. However, Lean startup thinking has not become a mainstream phenomenon even among the younger professionals. D's CEO says that some old entrepreneurs, who have never even implemented Lean methods in their companies, might be using the term on paper and speeches just because it has a modern and trendy image. This could make the term better-known, but also cause confusion about its meaning.

Startup A's CEO finds it very burdensome that the Lean startup thinking is not recognized by those Finnish public agencies who give financial support to selected companies. Governmental agencies demand a traditional-style business model that predicts several years of the future. A's CEO calls this policy ridiculous because no entrepreneur can see that far into the future.

The Finn's scarce Lean startup familiarization is revealed also when the interviewees are asked about where their startups' Lean know-how comes from. Only one of the four says that the whole founding team was aware of Lean startup methods on the company's day one. It is more common that one of the founders has studied Lean startup materials and then shares his knowledge with the other key persons. Naturally, four interviews do not provide any quantitatively reliable results, but considering that

the interviewed individuals are experienced professionals, this kind of a negative result is worth mentioning.

When it comes to the disadvantages of the Lean startup concept, the entrepreneurs take their time to think. Startup B's founder gives the most diplomatic answer: "In startups everything is so unsure that I cannot point out any particular disadvantage in the concept. I can imagine that there are situations where implementing the concept is not reasonable... but no example comes to my mind right now."

Startup C's founder thinks that the Lean startup concept may overly emphasize pivots. Pivoting gets easily misunderstood: entrepreneurs might think that they need to pivot as soon as they face the first problem. Pivoting can look like an easy way out of those troublesome situations where entrepreneurs should rather focus on optimizing and polishing whatever they have been doing so far.

Software entrepreneurs say that adopting the Lean startup thinking was a natural decision for them. In fact, some of them had been working in a somewhat Lean-like way before they even discovered it as a documented, specified concept. That is no wonder – they knew about Agile techniques long before hearing about Lean philosophy.

The entrepreneurs say that their startups have concretely implemented the Lean startup elements, but they warn about going "too Lean" and doing everything by-the-book. Each startup is unique, and Lean startup is a framework rather than a law. Reading manuals without understanding how well the teachings fit your own startup leads to a situation where every singular detail in the company might be well-organized, but the big picture is lost.

3) General questions about the business model

The two previous sections provided the general background information that is required in order to understand the four startups' resources, current situations and their views on the Lean startup concept. The startups have already explained how they see Lean startup as an idea – now it is time to discuss how the four Lean startups see themselves.

All four startups sell their products in the business-to-business (B2B) manner, which makes the mutual comparison between them more reasonable. Since the products are A) tools for marketing analytics, B) cloud-based services for testing user interfaces, C) communication tools for software developer teams and D) retail analytics, it comes as no surprise that the customers are companies instead of individual consumers.

Section 3.2.1 underlined that before any sales can begin, a startup must ask itself, "What kind of a startup is this?" The market type influences everything that a startup does.

	Startup A	Startup B	Startup C	Startup D
Entering an existing market	"We have nothing radically new... on the other hand, we do have"	"We have taken automated testing into a cloud environment"		"We are disrupting the market with our new product"
Creating a new market			"We were limiting ourselves, but now our aim is to create a new market"	
Resegmenting with low cost				
Resegmenting as a niche player				
Cloning a foreign model				

Table 1. Researched startups' market types.

Table 1 is a presentation of the startups' market types. Out of the five possible market types, resegmenting an existing market with low cost and cloning a foreign business model would be the most straightforward tactics. Resegmenting an existing market as a niche player is a good plan for a startup that does not aim at outplaying any big companies. The observed startups, however, show more remarkable ambition: all of them are either entering an existing market or creating a new one. Or at least they see themselves that way at this point of their lifecycles.

When it comes to market types, startup A is the most difficult to categorize. If one takes A's CEO's words literally, the startup could also be seen as a creator of a new market: "We have nothing radically new... on the other hand, we do have. This is online marketing analytics. What is new is that we are building a new category of analytics." However, the "new category" is built with tools that are likely to rival competitors with features, at least for some time. In this work A is placed in the row of "entering an existing market", but it is possible that somewhere in the future A will end up creating its own market.

B has taken user interface testing into a cloud environment, which brings the benefit of having the service ready-to-use immediately, for no installation is required. Furthermore, B has detected that few customers have the know-how for automated testing, so B aims at providing these skills. D believes that the startup has more modern solutions than the rivals, so that the market will be disrupted. Both B and D expect to face significant competition in their markets. Both startups are aware that the markets are crowded, but the startups believe they can do some things better than anyone else. This is by no means an easy path to success, but because the market already exists, these startups at least know who their customers are.

Startup C's market type has been a subject to change, because the customer segment and the problem/solution fit took some time to be found. C's founder explained that in the beginning the startup didn't implement all the ideas that they had about their own product, and therefore they didn't have enough substance and credibility for creating a new market. In other words, the entrepreneurs were limiting themselves. Now they are combining the original product ideas with the existing product and aim at creating a new market. The product is a communication tool for teams, but the catch is to connect it with the team's activity stream, so that all team members know precisely what is

happening around them. This way team's internal communication gets connected to project management.

Every startup is running at least one of the three engines of growth. The engines are called sticky, viral and paid, as explained in Section 3.1.3. Different engines of growth have differing importance to startups: if, for example, a startup is focusing on project-oriented business where customerships are short-lasting, the viral engine of growth is more suitable than the sticky one.

"Getting out of the building" to find customers and get meaningful feedback is an important action in the Lean philosophy. The four researched startups agreed that the entrepreneurs cannot focus exclusively on the product and expect customers to find it on their own. A's CEO summarized it: "If a company's marketing strategy is solely built on grapevine, the staff might as well quit and go home." However, this statement does not mean that gossips never matter. They do matter, but a company needs to make an effort to create and spread positive gossips about its own product.

Before doing the interviews with the four startups, the writer of this work had the following preassumptions:

1) If a startup is running the sticky engine of growth, there is a good chance that it is not utilizing the viral engine, and vice versa. These two strategies are so different.

2) If a startup is running the viral engine of growth, there is a good chance that it is using at least some money for paid advertisements. The paid and viral engines of growth are easy to run simultaneously.

3) Having the paid engine of growth as the only engine is too expensive for most startups.

Table 2 shows the four startups' engines of growth. The results clearly support the three pre-assumptions. A and D are mainly focusing on long-lasting customerships and do not trust in the viral visibility and the profitability of paid advertisements. B and C rely more on the combination of viral growth and paid advertisements.

	Startup A	Startup B	Startup C	Startup D
Sticky	"If customers pay once and then leave, it implies that they are disappointed"			"Long-lasting customerships are extremely important to us"
Viral		"We spread our message actively on Internet forums and IRC"	"Integration is very important"	
Paid		"Buying search engine visibility can get you unexpected customers"	"We have online ads, but those are unreasonably expensive"	

Table 2. Researched startups' engines of growth.

The choice of engines proved to play an important role in the startups' actions. B's founder proactively dedicates time for the startup's online presence by spreading the word about B's product on several Internet forums and Internet Relay Chat (IRC). B has also spent money to get better visibility in online search engines. According to B's founder this is a good way to attract customers whom you otherwise would not have identified as a part of the customer segment.

C's viral growth is particularly interesting because it is happening in two different ways. First of all, C's product (communication tool for software developer teams) is integrated with 40 other services. These services mention C's product in their own blogs and newsletters. This has proved to be a significantly important way for C to acquire new customers. Secondly, many of C's customers are companies that are growing rapidly and have multiple software developer teams. When one team in a customer company

tries out C's product, the product will likely get adopted by other teams, too. Occasionally C has also paid for advertisements, but the founder has come to find that unreasonably expensive compared to the profit.

In almost any company, it is an easy mistake to believe that the product is the epicentre of doing business. What is even more important is the business model; one could say that business model equals business. In a Lean software startup, understanding the Lean startup philosophy is therefore required for building a profitable business model. All four interviewed entrepreneurs are regular readers of books, blogs and other sources of information that can improve their understanding about the Lean startup concept. Two of the four entrepreneurs emphasized the importance of reading recent blog entries about the topic, for many blogs are updated frequently and they contain more recent ideas than printed books.

Some influential Lean books were criticized in the interviews, most notably *Lean Startup* by Eric Ries and *The Four Steps to the Epiphany* by Steve Blank. A's CEO called both of these well-known books "painful to read" and wondered who follows those. C's CEO considered the 2005 book *The Four Steps to the Epiphany* partially outdated. More recent publications, for example Ash Maurya's *Running Lean* received more positive feedback from the entrepreneurs. However, they believe that not even the best Lean books can be directly implemented in the real world without any adapting and interpreting. In a philosophy where uncertainty is taken for granted, there cannot be an absolutely definite how-to manual. Successful entrepreneurship cannot be achieved in a brain-dead way. A's CEO puts it well: "If you try to use some idea without adapting it, you will most likely fail. The whole point is to understand *why* are you doing what you are doing. If you have really understood the idea you read about, then you will adapt it in your own business almost automatically."

A direct consequence from the lack of perfectly guaranteed how-to manuals is that entrepreneurship experience is irreplaceable. Certainly one can succeed with his first startup, but experience is an advantage that cannot be studied from books, because the real-life cases will not happen by-the-book. D's CEO: "For the students I want to highlight that you must recognize your own unawareness. Only the reality can teach what the reality is like. You must adapt and interpret and read between the lines."

Although books, blogs and other educational material cannot guarantee a startup's success, they do provide many useful methods for building a business model. Literature provides tools, but entrepreneurs must learn by themselves how to use those tools.

The interviewed entrepreneurs are familiar with the Business Model Canvas (Section 3.2.4) and the Lean Canvas (3.2.5). Table 3 shows which startups use or have used these tools.

	Startup A	Startup B	Startup C	Startup D
BMC		"We have been using the BMC"		"Any simple tool is better than a 20-page-long business plan"
LC	"We are testing it right now"	"We have just started to use the LC"	"Our staff gathers once a year to fill the LC"	

Table 3. Researched startups' usage of the BMC and the LC.

The entrepreneurs saw the BMC as a valuable drafting tool, especially for beginners. D's CEO: "Once you have truly understood the BMC, you might not have to write it down, or you can replace the BMC with mind-maps, PowerPoint slides and other tools like that. Any simple tool is better than a 20-page-long business plan. You'll throw that away sooner or later anyway."

When the interviews were done in May and June 2013, the Lean Canvas was becoming better-known and more popular. Both A and B had recently started using it. B's founder admitted that he started using the LC only because it was recommended to him and he was unable to say which canvas is more useful.

The two canvases can be used for both internal and external communication. C's CEO: "Both of them are useful communication tools. Our staff gathers once a year to fill the

LC together, so that everyone knows the business plan and understands where the is coming from. [...] And when the canvas is filled, we use it to explain our business to outsiders." The CEO's positive words contradict with startup C's stark reality: if the staff updates the Lean Canvas *once a year*, it is not seen as a useful communication tool.

None of the four startups have fixed and scheduled Learning Loop iterations (Section 3.1.5). The Learning Loop is acknowledged in the startups, but its usage is rather an emerging result of other actions than an action of its own. When the software products go through changes, startups gather feedback from customers and learn from that, but there are no fixed iterations in any of the four startups.

4) Customer Discovery

In Lean startup literature, Customer Discovery precedes Customer Validation. Customer Discovery starts with the entrepreneurs' vision which is then turned into business model hypotheses. Customer Validation tests whether the entrepreneurs' plan of doing business is repeatable and scalable. Should it fail the test, it needs to be pivoted, so the process steps back to Customer Discovery.

Reality differs from literature. All interviews revealed that the researched startups have been operating Customer Discovery and Customer Validation simultaneously. Entrepreneurs justified it with arguments like "we knew the customers' problem on the startup's day one" or "our startup is a quick player, so we sell the product and ask for feedback at the same time".

While Customer Discovery has been going in, startups have had a number of hypotheses: "Can we solve this customer segment's problems?", "Are customers willing to pay for this software?" and so on. The Lean startup approach for validating these hypotheses is getting early feedback from potential customers. All of the four entrepreneurs have received valuable results from customer feedback. Eventually they all learned that there were more unknowns than they had first thought. D's CEO: "We thought we knew everything, but things turned out to be hypothetical. The only fact that we actually knew was that there is a growing market for our kind of a product." B's

founder: “On B’s day one we were not very informed about the Lean startup concept and we had not written down any hypotheses. Once we adopted the Lean startup philosophy, we started to talk with customers much more than before. That is when we realized that we do not understand the customers’ problems. The Lean startup thinking has made us more humble: we do not know everything, so we must talk to the customers.”

Listening to customer feedback is a good example of an action that requires more practical skills than what the books can teach. A customer might be struggling with unknown issues just as badly as a startup. Startup A’s CEO: “I followed the teachings of the book *Running Lean*, so I wrote three important business hypotheses on a piece of paper. Then I made iterations: if one hypothesis proved to be wrong, I replaced it with a new one. I asked customers to put my hypotheses into order by their significance. I learned that how the customers list my hypotheses is *less* important that what they are saying while they are doing that. It would often happen that a customer talks about some problem for quite a while but then marks it less important on the paper. Luckily I have been in this business for a long time, so I was able to understand the customers’ behaviour.”

Because the startups have been doing Customer Discovery and Customer Validation simultaneously, they have also used revenue as a key metric since very early days. Looking at the income stream right away seems to have been worthwhile because it made the entrepreneurs to concentrate on the value their product has. D’s CEO: “In the very beginning we failed to realize what is valuable to the customers. We realized our mistake only when there was no revenue coming in!” A’s CEO: “Every metric except revenue is a vanity metric for us at this point. Other things matter when you focus on optimizing.” B’s founder: “I find revenue the most important number... but in the future we’ll pay increasingly attention to the number of active users.”

The entrepreneurs have very similar plans for acquiring new customers and keeping them active. Their methods are familiar from almost any Lean startup manual: marketing the product to new customers with phone calls and on LinkedIn, explaining why the product is valuable and sending newsletters and notifications to existing customers. Startup C provides the only differing answer, for it aims at acquiring new

customers by integrating the product with many other services. This is an expected answer because of C's viral engine of growth.

One important part of Customer Discovery is releasing an MVP quickly, as explained in Section 3.1.1. Early MVPs are used to test assumptions and minimize the lost resources. The most usual MVP-related mistake is to add too many features, i.e. more than are needed for testing assumptions. Three of the four interviewed entrepreneurs made this mistake in their startups, and in retrospect they do realize that it was a mistake. They fell for perfectionism with the MVPs, which unnecessarily cost them time and effort. The only startup that released an MVP very quickly was satisfied with the learning they got through that MVP.

The purpose of Customer Discovery is finding the problem/solution fit. MVPs and other validation methods help to discover it little by little, but already in the early days a startup should know at least one of the these three variables: customer's problem, the solution or the customer segment. When one of these is fixed, there is a reasonable direction to search for the remaining two. If nothing but the development technology is thought about on a startup's day one, it is not Lean startup development. D's CEO knows this from his own experience: "At first we knew what technology we are going to use. Nothing more than that. It was very Finnish: First you have the technology, then you start searching for a problem to be solved. When we started working in the frames of Customer Development, everything changed. Now we start by identifying problems and then we look for solutions."

It is a long way to go when a startup is looking for a profitable business model, but the time spent in the process is not wasted if the learning gets validated. This is arguably one of the main advantages of Lean startup thinking. It is highlighted in the literature and it gets credit in real, too.

While the business model is discovered bit by bit, the learning that happens is validated. This is arguably one of the main advantages of Lean startup thinking. Validated learning is highlighted in the literature and it gets credited in real life, too. D's CEO: "If we had simply guessed everything correctly on the startup's day one, we would not have learned those things that we have now learned. I strongly believe that mistakes

teach better than anything else. Our team has young people with little experience... if mistakes can teach people, then we have definitely learned a lot!"

5) Customer Validation

In the Customer Validation phase, a startup should make its business model repeatable and scalable. This requires that the startup acknowledges whom the product is made for, and how the startup differs from the possible rivals. Realizing the market type, engine of growth and other previously discussed issues are some of the early steps on the path towards succesful Customer Validation.

Customer Validation may begin when the first versions of the deliverables from the earlier phase, namely Customer Discovery, exist. These deliverables are a problem statement document, a product requirement document, a revenue plan, and solid business and product plans. Much like everything in Lean startups, the content of these documents is a subject to change. Doing Customer Discovery and Customer Validation at the same time is not a valid reason to leave any of these undone. However, the four entrepreneurs were not interested in writing problem statement and product requirement documents down. Such documents were seen as unnecessary. D's CEO: "You need those documentations just so that you can show those to the investors. We have minimized this kind of documentation. Then again, it could be a good piece of practice and training to write those documents, although the entrepreneur would never use those." When an entrepreneur has experience from previous companies, he may start thinking that writing statements and plans down is for beginners. This is not the attitude that the Lean startup literature commends.

The entrepreneurs claimed both explicitly and implicitly that they know the essential elements of their Customer Validation, such as whom the product is made for and what are the assets of the startup, quite well. This kind of self-assurance is surprising in the Lean startup world where uncertainty should be taken for granted.

What makes the entrepreneurs' self-assurance even more surprising is the fact that they are aware of their past mistakes. Being aware of the world's uncertainty should be an

active and disciplined process. Otherwise entrepreneurs may fall for unjustified tranquility.

Interviews revealed that the ways how the four startups see themselves, as well as what the startups are doing, have changed during their existence. A's CEO: "Our Customer Segments have changed, as well as the pricing. Furthermore, the way how we tell about our product to the outside world has changed three times." B's founder: "It was a big change when we included know-how into the service: instead of selling just a tool, we'll also sell a service where someone with good skills is using the tool for the customer's benefit." C's CEO: "At first we were planning to create a work logging tool, but it evolved into an increasingly social product. Eventually it became a team communication tool, so we dropped the hour logging feature."

In other words, the startups have pivoted. Interviews revealed that some of the researched startups have fixed cyclic times when they proactively think about pivoting, while other startups see pivoting as an intuitive idea. D's CEO: "Our board holds a meeting once a month and in these meetings they always think about the startup's direction." B's founder: "When we radically change something in our business plan, it usually just pops up in mind. It is like an unplanned storm in the head." The latter attitude towards pivoting is undoubtedly challenging to the beginning entrepreneurs, because good intuition often requires experience. The importance of experience is also mentioned by A's CEO: "I have been doing this for so long that I can tell the normal doubts from that anxiety which occurs when there's something seriously wrong."

Pivoting was seen as a redirection towards the startup's vision, not as a way to change that vision. A's CEO: "Pivoting gives you an alternative path to your vision, so if you do not have a vision – and it looks like many startups do not – then pivoting is extremely difficult. You can get it right with pure luck, but if you know what you want to achieve, things are easier: you pivot so that you can avoid problems by choosing another path to your goal." The quote is true when the original vision is both achievable and profitable – in other cases pivoting is a way to either improve the original vision or fail fast.

While the fundamental meaning of pivoting can be argued, its importance is clear. D's CEO: "During 2013 we have grown from five people to eight people, and we are going

to nearly double it by the end of the year. This growth is all thanks to successful pivoting. Without pivots we would be in bankruptcy."

The making of the decision to either pivot or persevere should be supported by customer feedback and user analytics. All of the four entrepreneurs underlined the importance of customer feedback, as expected from Lean software startups. The importance of analyzing the feedback properly was also acknowledged by every interviewee. B's founder: "It would be dangerous to simply ask if the customer is satisfied, because some customers might be overly polite. In customer interviews we always have two of our staff members; both of them make their own notes and after the interview they form a synthesis of their individual notes. In this month we are going to do dozens of these face-to-face customer interviews." C's CEO: "It takes our developer team lot of effort to analyze the customer feedback, but it is worth it. Every person in our team can see the feedback. Perhaps this creates subconscious understanding about the customers' preferences."

Pivoting is a trendy action among the Lean startup community. While pivoting can be a reasonable decision, there is the risk that an entrepreneur pivots too hastily. D's CEO: "At the moment, there are people who want to pivot no matter what. However, the old business model should be properly tested before that. One should get lots of customer feedback and also look in the mirror every now and then. The business idea might be better than its executer. It takes some intuition to realize this."

6) Miscellaneous

When the entrepreneurs were asked if they are satisfied with the Lean startup concept and if they would use it again, none of them was dissatisfied, but doing things "by the book" was not advocated by any of them. B's founder: "In my previous company we did lots of unnecessary work. The Lean philosophy is much more reasonable... I have not found anything better than the Lean startup concept." C's CEO: "The Lean startup concept includes many good ideas, but there's the risk that one misunderstands the context and the scope. I do not do things by the book; I rather read other people's thoughts and then conclude what is reasonable." D's CEO: "Everyone should be aware

of his own ignorance. Being a Lean startup entrepreneur cannot be learned from the books. Besides, every book is somewhat outdated."

Being aware of one's own ignorance was seen as the very core of Lean startup thinking. B's founder: "What makes our startup a Lean startup is that we have several hypotheses which are known unknowns and we are trying to validate those. We do not claim to know anything that we do not know. I think this is the core of Lean startup thinking."

6 Summary and Conclusions

With or without Lean thinking, most software startups fail. Lean startup philosophy is not designed to be a magic bullet that always hits its target. It merely helps the entrepreneurs to test their business hypotheses, validate the learning and fail fast (rather than slowly) in a world of uncertainty. There are two types of uncertainty: product-related and business-related. Products can be tested by using prototypes, whilst business models can be tested by utilizing the Lean startup principles.

Lean startup thinking is explained in several books that are written by experienced entrepreneurs. Some of those are written in a step-by-step way that might lure an unexperienced reader to think that being a software entrepreneur is easy when you just follow the instructions. This attitude is dangerous and should be avoided. Experience is indispensable. There are no two identical startups. The four interviews that were done for this thesis showed that there are Lean software startup entrepreneurs who are very sceptical towards following "startup owner's manuals". One of them said it like this: "When I was a student, I wrote my Master's thesis on Lean startups. Back then I thought that I know a lot of those, but when I became an entrepreneur I realized that I know next to nothing. There are so many variables... many things depend on what kind of a team you have and what is going on in the market. Studying from books is not a bad thing, but that knowledge should be seen in the right context." Indeed, any non-fictional book is an interpretation of some limited part of reality.

Even though only four Lean software startup entrepreneurs from four different companies were interviewed for this thesis, it was enough to show that there are different ways of adapting Lean startup philosophy in software startups – and the real-life cases differ from the books' instructions. However, the interviewees found the philosophy valuable and claimed that they would use it again in another startup. Because of their experience they understood that business cannot be done "by the book". Step-by-step Lean startup manuals are a form of productizing a philosophy which should be implemented differently in different cases.

Being flexible is one of the main goals of Lean startup philosophy. The entrepreneurs can take this so far that they are flexible in the process of implementing the philosophy – they may highlight some Lean startup elements more than the others, and leave some elements completely out of their working routines. This is reasonable as long as it results in successful high-performance processes. On the other side, entrepreneurs might cheat themselves to believe that they are actually using certain Lean startup tools, for example by filling the Lean Canvas once a year or so.

The literature underlines the importance of iteration cycles, but none of the four contacted companies had a clear cyclic way of working. Even when a startup consists of less than ten people, performing planned cycles is a serious challenge.

There are several metrics and tests that are considered essential for early startups, but the interviewees saw the contemporary revenue much more important than any other metric. That is a myopic way of thinking, but it *can* work. In general, real Lean startups implement the philosophy *ad hoc* and thus become increasingly different from each other. The philosophy seems to be mostly correct, but there is no startup that could prove every single part of it. Lean startup thinking is a collection of tools, and one can succeed in his building project without using every tool in the toolbox.

The Lean startup philosophy is spreading rapidly and supported by an increasing amount of literature. Some of the most influential Lean startup books were published just shortly before the making of this thesis, or even during the writing process. Therefore, Lean startup is a concept that is still looking for its own shape. How entrepreneurs understand Lean startup thinking and how they implement it is a subject to change.

References

[1] Graham, Paul. “Startup = Growth” September 2012. <http://www.paulgraham.com/growth.html> Accessed 4 March 2013.

[2] Poppendieck, Mary & Tom. 2003. *Lean Software Development: An Agile Toolkit*. Addison-Wesley Professional. ISBN: 978-0321150783.

[3] Difference Between.com. “Difference Between Lean and Agile” 22 April 2011. <http://www.differencebetween.com/difference-between-lean-and-vs-agile/> Accessed 8 March 2013.

[4] Liker, Jeffrey. 2004. *The Toyota Way – 14 Management Principles from the World’s Greatest Manufacturer*. New York, NY, USA. ISBN: 0-07-139231-9.

[5] Kellogg School of Management. “Toyota Production System” <http://www.kellogg.northwestern.edu/course/opns430/modules/lean_operations/ohno-tps.pdf> Accessed 24 March 2012.

[6] Saito, Akinori. ”Message from Mr. Saito, President of TSSC” 2009. <http://www.tssc.com/welcome.htm> Accessed 24 March 2012.

[7] Fried, Jason & Hansson, David. 2010. *ReWork*. Chatham, UK: Vermilion. ISBN: 9780091929787.

[8] Schultz, Rick. ”James Cameron tells the astonishing story of Titanic, his breathtaking labor of love” 1997. <http://www.industrycentral.net/director_interviews/JC01.HTM> Accessed 30 November 2012.

[9] Highsmith, Jim. ”Beyond Scope, Schedule, and Cost: The Agile Triangle” 14 November 2010. <http://jimhighsmith.com/2010/11/14/beyond-scope-schedule-and-cost-the-agile-triangle/> Accessed 30 November 2012.

[10] Moran, Dave. ”Where Do Unused Features Come From?” 12 July 2011. <http://www.softwareresults.us/2011/07/where-do-unused-features-come-from.html> Accessed 1 December 2012.

[11] Lochan, Rupesh. “Lean or Agile? A Comparison of Approach” 8 April 2011. <http://www.processexcellencenetwork.com/lean/articles/using-lean-in-agile-software-development-a-compari/> Accessed 7 January 2013.

[12] Jul, Martin. ”The Five Whys of Lean” 20 February 2007. <http://community.ative.dk/blogs/ative/archive/2007/02/20/The-Five-Whys-of-Lean.aspx> Accessed 24 April 2012.

[13] Ries, Eric. ”How to conduct a Five Whys root cause analysis” 2 July 2009. <http://www.startuplessonslearned.com/2009/07/how-to-conduct-five-whys-root-cause.html> Accessed 24 April 2012.

[14] Lean Manufacturing Shop. ”Implementing Lean Manufacturing” 2011. <http://leanmanufacturingshop.com/implementing-lean> Accessed 12 February 2012.

[15] Ries, Eric. ”The Lean Startup: Innovation through experimentation” 3 November 2010. <http://www.slideshare.net/500startups/eric-ries> Accessed 10 February 2012.

[16] Lean Startup Machine. “Lean Startup Machine: Frequently Asked Questions” 2012. <http://leanstartupmachine.com/faqs/> Accessed 8 March 2013.

[17] Sokullu, Emre. ”How To Create A Minimum Viable Product” 13 July 2012. <http://techcrunch.com/2012/07/13/how-to-create-a-minimum-viable-product/> Accessed 23 November 2012.

[18] Ries, Eric. 2011. *The Lean Startup*. New York, NY, USA: Crown Business. ISBN: 978-0-307-88789-4.

[19] Ries, Eric. “Validated Learning” <http://lean.st/principles/validated-learning> Accessed 10 January 2013.

[20] Lofgren, Lars. “The Three Engines of Growth – with Eric Ries” <http://larslofgren.com/marketingbasics/the-three-engines-of-growth-with-eric-ries> Accessed 3 February 2013.

[21] Cooper, Brant & Vlaskovits, Patrick. 2010. *The Entrepreneur’s Guide to Customer Development*. Breinigsville, PA, USA: CustDev. ISBN-13: 978-0-9827436-0-7.

[22] Robbins, Zach. “Lean Startups: Part 1 – ‘Learn, Measure, Build’” 27 July 2011. <http://viget.com/advance/lean-startups-part-1-learn-measure-build> Accessed 10 January 2013.

[23] Osterwalder, Alexander & Pigneur, Yves. 2010. *Business Model Generation*. Hoboken, NJ, USA: John Wiley & Sons, Inc. ISBN: 978-0470-87641-1.

[24] Blank, Steve. 2005. *The Four Steps to the Epiphany – Successful Strategies for Products that Win*. Cafepress.com. ISBN: 978-0976470700.

[25] Marcus, Jonathan. “High Definition – What is up now!” 17 October 2007. <http://vimeo.com/blog/post:70> Accessed 24 January 2013.

[26] Lowensohn, Josh. “YouTube quietly launches official support for HD” 5 December 2008. <http://news.cnet.com/youtube-quietly-launches-official-support-for-hd/> Accessed 24 January 2013.

[27] Where 2 Get It, Inc. “In-N-Out BURGER: All Locations” <http://hosted.where2getit.com/innout/newdesign/all-locations.html> Accessed 2 February 2013.

[28] Blank, Steve & Dorf, Bob. 2012. *The Startup Owner’s Manual – The Step-by-Step Guide for Building a Great Company*. USA: K&S Ranch Publishing Division. ISBN: 978-0-9849993-0-9.

[29] Maurya, Ash. 2012. *Running Lean – Iterate from Plan A to a Plan That Works*. USA: O’Reilly Media, Inc. ISBN: 978-1-449-30517-8.

[30] Nonlinear Thinking. “The Business Model Canvas” 5 July 2008. <http://nonlinearthinking.typepad.com/nonlinear_thinking/2008/07/the-business-model-canvas.html> Accessed 12 March 2013.

[31] Business Model Foundry. “The Business Model Canvas: Your business model – on one page”. <http://www.businessmodelgeneration.com/canvas> Accessed 12 March 2013.

[32] Osterwalder, Alexander. “The Business Model Theater – Can You Put on a Show?” 31 January 2013. <http://www.businessmodelalchemist.com/2013/01/the-business-model-theater-can-you-put-on-a-show.html> Accessed 12 March 2013.

[33] Maurya, Ash. “Lean Canvas – How I Document my Business Model” August 2010. <http://www.ashmaurya.com/2010/08/businessmodelcanvas> Accessed 4 April 2013.

[34] Maurya, Ash. “Why Lean Canvas vs Business Model Canvas” 27 February 2012. <http://leancanvas.com/blog/why-lean-canvas-vs-business-model-canvas> Accessed 4 April 2013.

[35] Lofgren, Lars. “Metrics, Metrics On The Wall, Who’s The Vainest Of Them All?” 2012. <http://blog.kissmetrics.com/vainest-metrics> Accessed 9 March 2013.

[36] Bales, Scott. “Lean Startup – Funnel Analytics” 10 April 2013. <http://www.slideshare.net/scottiebales1/slsc-funnel-analytics-scott-bales> Accessed 16 May 2013.