MASTER THESIS

TITLE: Lean management applied to new business development: experts’ analysis and examples of hardware companies

MASTER DEGREE: Master in Science in Telecommunication Engineering & Management

AUTHOR: António Pedro Palma Santos Gomes

DIRECTOR: Jordi Olivella

DATE: January, 8th 2013
Títol: Lean Management Aplicat al Desenvolupament de Nous Negocis: Anàlisi dels experts i exemples d'empreses de hardware

Autor: António Pedro Palma Santos Gomes

Director: Jordi Olivella

Data: 8 gener 2013

Resum

Aplicacions amb molt d'eixit del Lean Startup han anat sorgint últimament en la indústria del sofware, sobretot de startups als EUA. A més, recentment ha estat utilitzat pel govern dels Estats Units que també han adoptat el mètode en el desenvolupament d'un servei públic en línia.

L'objectiu principal d'aquest escrit ha estat analitzar si l'enfocament científic proporcionat per la metodologia Lean Startup és aplicable per al desenvolupament de hardware. Per arribar a aquestes conclusions, es va buscar l'experiència dels empresaris que són els propietaris de les Startups que estan en vies de desenvolupament de productes tecnològics.

El llibre d'Eric Ries "Lean Startup" va ser considerat, en contes d'altres contribucions importants, com per exemple, "Running Lean" per Ash Maurya. Moltes de les fonts consultades són molt recents, donat que encara esta en funcionament i va canviant.

Es van realitzar entrevistes a empresaris de les empreses: Alteraid, Fractus, On-Sun Systems i Technology Assistance BCNA. Les entrevistes van consistir en una primera presentació breu dels conceptes principals de la metodologia, i un debat al voltant de 50 minuts sobre el tema de la possible aplicació de l'enfocament Lean Startup per a les seves noves empreses, en base als seus coneixements i experiència adquirida en el desenvolupament dels seus negocis. Les entrevistes van consistir un grup de 15 preguntes.

Totes les opinions recollides indiquen que Lean Startup és aplicable al desenvolupament de hardware. L'eliminació de problemes (temps i diners), l'agilitat de desenvolupament de productes, així com la interactivitat client, sembla aplicable, adequada i avantatjosa per al desenvolupament de productes físics es a dir hardware. No obstant això, per tal d'aplicar el mètode per al desenvolupament de maquinaria es requerirà l'adaptació, perquè els resultats suggereixen que és dependent de la indústria.

Paraules Clau
Lean Startup, Agilitat, Eficiència, Eficàcia
Overview

Very successful applications of the Lean Startup’s principles in the software industry have been emerging lately, especially from startups in the USA. Additionally, it has recently been used by the American government who have also adopted the method in the development of an online-based public service.

The main goal of this dissertation was to analyse whether the scientific approach provided by the Lean Startup methodology is applicable, as well as a plus, to hardware development. In order to achieve such conclusions, it was sought the experience of entrepreneurs who own startups developing (physical) technological products.

The Eric Ries’ book “Lean Startup” was considered in besides of other important contributions, e.g. “Running Lean” by Ash Maurya. Several of the sources used were very recent given the still on-going movement around the subject.

Interviews were conducted to entrepreneurs from the companies: Alteraid, Fractus, On-Sun Systems and Technology Assistance BCNA. The interviews consisted in a first brief presentation of the methodology’s main concepts, and a 50 minutes debate around the subject of the possible applicability of the Lean Startup approach to their startups, based on their knowledge and experience gained developing their businesses. The interviews covered a group of 15 questions.

All the entrepreneurs’ opinions from the interviews indicate that the Lean Startup is applicable to hardware development. The removal of wastes (time and money), product development agility, as well as customer interactivity, seem to be applicable, suitable and advantageous to the development of physical products. However, in order to apply the method to hardware development it will require adaptation, which the results suggest that it is industry dependent.

Key Words
Lean, Startup, Agility, Efficiency, Efficacy
I would like to thank Professor Jordi Olivella, from EETAC, for offering this interesting theme to work on and for the support given.

I am also thankful to Professor António Rodrigues, from IST, for his aid in the process of developing this work.

Last but not the least, also a sincere thanks to those that directly or indirectly help me fulfilling this project.
“Most of all, we would stop wasting people’s time.” (Eric Ries)
INDEX

INTRODUCTION .................................................................................................................. 1

CHAPTER 1. BEFORE LEAN STARTUP ................................................................. 2
1.1. Strategic Planning, Entrepreneurial Orientation and Startups .............................................. 2
1.2. The Emergence of the Agile Approach .................................................................................. 5
1.3. Agile Vs. Lean Startup Methodologies ................................................................................. 7
1.4. The Lean Startup Approach beyond “Fast Industries” ......................................................... 9
1.5. The “Lean” Origins .......................................................................................................... 10

CHAPTER 2. THE LEAN STARTUP APPROACH ........................................ 12
2.1. Introduction ..................................................................................................................... 12
2.2. Startups: Why Lean? ....................................................................................................... 13
2.3. Progress Unit: Learning ................................................................................................... 15
2.4. Becoming a Lean Startup ................................................................................................. 16
   2.4.1. Making Progress ....................................................................................................... 19
   2.4.2. “First-version” Products: MVPs .............................................................................. 20
   2.4.3. Measuring Progress: Innovation Accounting and Cohort Analysis ...................... 23
   2.4.4. Maintaining or Changing Strategy ......................................................................... 28
   2.4.5. Small Batches and Continuous Deployment ............................................................ 29
   2.4.6. Customer Development .......................................................................................... 32
2.5. Boosting the Engine of Growth ........................................................................................ 34
2.6. Summary of Steps to Take .............................................................................................. 35

CHAPTER 3. INTERVIEWS ........................................................................... 37
3.1. Questions ......................................................................................................................... 37
3.2. Interview Results ............................................................................................................ 38
   3.2.1. Alteraid Interview .................................................................................................... 39
   3.2.2. Fractus Interview .................................................................................................... 43
   3.2.3. On-Sun Systems Interview ..................................................................................... 47
   3.2.4. Technology Assistance BCNA 2010 Interview ....................................................... 51

CHAPTER 4. CONCLUSIONS ......................................................................... 56
4.1. Findings ........................................................................................................................... 56

CHAPTER 5. BIBLIOGRAPHY ....................................................................... 64
CHAPTER 6. APPENDIX .................................................................................. 66

6.1. Adapted strategic planning model S_EWOET matrix ........................................ 66
6.2. Strategic Process Variables ........................................................................ 66
6.3. The Lean Principles ..................................................................................... 67
6.4. Traditional vs. Lean Startup Comparison Summary ....................................... 71
6.5. The AARRR Metrics .................................................................................. 72
6.6. Prioritising Risk ......................................................................................... 73

Figures Index

Fig. 1.1 Benefits of the Agile development approach .............................................. 7
Fig. 2.1 Vision-Strategy-Product pyramid ............................................................. 15
Fig. 2.2 Three stages of a startup ......................................................................... 18
Fig. 2.3 The one-page business model “Lean Canvas” ........................................... 19
Fig. 2.4 Solution and Problem spaces .................................................................. 20
Fig. 2.5 Diffusion of Innovation ......................................................................... 21
Fig. 2.6 The 80/20 rule of thumb for prioritizing focus when building MVPs; 80% of the team's effort should be spent toward optimizing existing features ......................... 21
Fig. 2.7 The Build-Measure-Learn feedback loop ................................................. 23
Fig. 2.8 The traditional hockey stick graph: Vanity metrics ................................... 25
Fig. 2.9 Typical funnel report on a month period for a downloadable product .......... 26
Fig. 2.10 Weekly funnel cohort by join date ......................................................... 26
Fig. 2.11 Another example of a cohort analysis which clearly shows trends of each property in analysis, i.e. each cohort .................................................................................. 27
Fig. 2.12 Traditional product development; unit of progress: advance to next stage ........ 30
Fig. 2.13 Continuous Deployment Principles ...................................................... 31
Fig. 2.14 Continuous deployment cycle ................................................................. 32
Fig. 2.15 Customer Development diagram .......................................................... 33
Fig. 2.16 Steps summarizing the Lean Startup approach ....................................... 36
Fig. 2.17 Workflow to follow when building a Lean Startup ................................ 36
Fig. 4.1 Sequential versus iterative development .................................................. 63
Fig. 6.1 SEWOT matrix analysis...........................................................................................................67
Fig. 6.2 Pirate metrics....................................................................................................................................73
Fig. 6.3 Systematically tackling risk in stages..........................................................................................75
INTRODUCTION

This dissertation gives continuation to the, still on-going and very current, Lean Startup movement that has transformed and will continue to transform how new products are developed. The movement has continuously been receiving improvements and new contributions from all over the globe. As such, this work tries to be as updated as possible, by taking advantage of some of the latest available aforementioned contributions, for the purpose that it serves.

The purpose of this work is to find out about the viability of applying the Lean Startup principles and techniques in startups whose business products are not software- or web-based, or at least do not have those as product main attributes. In order to achieve that goal, the concepts and techniques of the Lean Startup methodology that have been applied largely in software products are overviewed. Afterwards, entrepreneurs from 4 different Startups are interviewed, in order to give their valuable opinion on whether those principles and techniques would be applicable and useful in their cases.

Regarding the environmental impact of the work, it is most probably great and positive. As the ultimate goal of becoming more Lean is to eliminate wasteful activities, which inevitably translates into energy savings once those wasteful activities are diminished or eliminated, and setbacks are avoided. By being able to apply the Lean Startup (scientific) approach to hardware development startups, on a fundamental level, less time and resources would be consumed by startups in the pursuit of (true) success.

The work is organized in 4 chapters: the first chapter gives a quick overview of the most common and current context in startups’ development, regarding some of the most common traditional management techniques, as well as the modern entrepreneurship approach that had emerged as a response to today’s turbulent and competitive world, and also an overview on the Lean origins.

Chapter 2 presents and explains the concepts and techniques of the Lean Startup methodology, from the point of view of software development, and which are worth mentioning for the purpose that this work serves.

Chapter 3 presents the interviews made to 4 entrepreneurs and the results of the debates conducted with each one, regarding the possible applicability of the Lean Startup approach to their startups, based on their knowledge and experience gained developing their businesses, covering a group of 15 questions.

And finally, chapter 4 lists the findings and conclusions regarding how much applicable and advantageous is the Lean Startup in hardware development.
CHAPTER 1. BEFORE LEAN STARTUP

1.1. Strategic Planning, Entrepreneurial Orientation and Startups

In today’s turbulent and competitive world, one cannot simply decide to build a business—a Startup—just because there is an excellent idea at stake, or even let the “flow” and luck drive his/her business metrics. The technology has progressed at such level that practically everything is possible to be developed, whereby the question becomes not if it can be built, but rather should it be built.

Strategic planning has become one of the most popular tools to adjust the goals and mission for the success of a business plan [4]. A strategic planning model can greatly reduce the required management knowledge needed for strategic planning and may enable startups to avoid problems such as complexity, time-consuming process and the need for skilled experts (ibid.). However, it is not immediately clear whether strategic planning can be used in startups.

Among all strategic planning models proposed along last two to three decades, all of them seem to have three parts in common: the mission statement, internal analysis for recognizing the strength and weaknesses and external analysis for using opportunities and avoiding threats. Although it is to expect that the basic strategic planning model used by very small organizations with no experience in strategic planning should also be applicable to startups, it is a wrong supposition; startups have some exclusive properties which differentiate them from small businesses [4].

The basic strategic planning model consists of five steps:

1) Setting goals and the mission statement;

2) Setting intermediary goals;

3) Setting tactics and strategies for achieving those goals;

4) Setting operational plans for implementing the strategies;

5) Revision and updating the plans.

The question was how to adapt this model in order to be used in startups.

One of the principles of the Lean Startup methodology is to view entrepreneurship as a different and unique type of management; this perspective begins to make more sense when one realises the typical challenges startups face while struggling to survive their very first months of “life” filled with uncertainty and risk. It is relevant to understand the important role of entrepreneurship in startups and even in already established firms. The main entrepreneurial characteristics include risk taking, creativity, locus of
control, need for achievement, diligence, visionary, ambiguity tolerance, and challenge ability. All these characteristics ultimately contribute to the expansion of the resources needed to create a new organization, therefore becoming obvious the value of entrepreneurship in building up a startup, whereby it makes sense that entrepreneurial concepts are, indeed, central in strategic planning for startups.

As result of a conducted case study on the Iranian packaging industry [4], emerges a proposal of a quite similar—but appropriate for startups—strategic planning model adapted for small businesses, differing only in the methodology used for internal and external analysis and in the parameters taken into consideration. The model is composed as follows: firstly the a) Vision, defined as the viewpoint and the founder’s beliefs towards the company’s internal issues, governance, competitive situation and its interaction with the environment; secondly the b) Mission, differing from the other methods by being solely based on the founder’s vision, it is defined as the skeleton for all the company’s strategic plans. These plans can be built by answering the questions: what are the company’s main activities and promises to the stakeholders? What are the products? Who are the customers and where geographically do the company’s activities end?

Afterwards, the c) Internal Analysis: used to find out about internal strengths and weaknesses where the company’s status (in terms of financial, cultural, management, marketing, production, R&D and IT) is evaluated. However, since a startup has no previous activity, many of these topics become irrelevant, therefore, Entrepreneurial Characteristics (founder’s creativity, innovation, risk tolerance, self-control, and intrinsic control, challenge acceptance, perseverance, etc.) and Internal Resources and Abilities (human and financial resources) are used instead to evaluate strengths. Still inside the Internal Analysis, are comprised the Competitive Advantages (which in a startup context consists in the production methods innovation, sales and marketing innovation, and organization innovation) and Weaknesses (which commonly in startups are: the lack of information systems and marketing department, the production system problems and an unshaped organizational structure).

On the other hand, the d) External Analysis is based on an environmental-opportunities-and-threats study focusing on micro environment characteristics (i.e. customers, suppliers, partners, competitors and financers), taking into consideration 4 main aspects for the analysis [4]:

- Entrepreneurial opportunities: the potential or real opportunities in the startup’s surroundings which can be used to reach its goals, and are mainly composed by the competitor’s weaknesses, customer needs and market’s state;

- Competitors and Market’s state: includes identifying main competitors and their strategies, evaluating their performance and market share, and analysing the market;
• External resources: identifying external resources and analysing them will open new opportunities, crucial for startups, resources that are not controlled by the company but can be used depending on the environmental conditions (social capital, banks’ or investors’ financial resources and human resources outside the company whose expertise can be utilized);

• Threats: different from the typical of established businesses, a startup’s threats comprise microenvironment elements like competitors and customers.

The last step of the proposed model is the e) Strategy and Operational Plans, which defines the type of strategy the startup will be based on; in this context, it can either be based on a focus strategy, a differentiation strategy, or a combination of both.¹

But we are living in times of great unpredictability which makes the value of planned strategies practically tend zero, because those strategies will often be based on flawed assumptions and inaccurate data. Even if the aforementioned strategic planning model for startups may be useful and valuable in that sense, namely by:

1. “Injecting” startups in the market as soon as possible by starting the planning process and reducing its time span through all the data gathering by the model;

2. Familiarizing the startup’s founders with scientific foundations of strategic planning and compelling them to the practice of strategic planning through the model’s simplicity, integrity and attention to special entrepreneurship characteristics;

3. Increasing startups’ success probability, guiding their founders in identifying competitive advantages, their goals, and entrepreneurial opportunities;

4. Attracting from the beginning, the founder’s attention to the macro and—mainly—micro environments. In the micro, they are able to actually make changes and have an active role;

5. Creating an infrastructure for future strategic planning and also a valuable database by collecting data from the very first days;

it still lacks a scientific method, supported by empirical data, that undoubtedly guides the startup towards success, i.e., a sustainable and prosperous business.

Also, what we have begun to witness a couple of years ago is a focus on processes and styles of strategy development—termed Entrepreneurial

¹ For the analyse of the adapted SWOT matrix see appendix 6.1
Orientation, EO—rather than the traditional focus and relevance given on content and results. The traditional view of entrepreneurship focuses on content rather than process, e.g., the individual entrepreneur—someone who took bold risks, made his/her way caring only about the achievement of his/her goals, relying on personal beliefs and hunches, not building anything that could be used again and improved each time, as long as he/she presented the desired results.

EO refers to the processes, actions, methods, practices, and decision-making styles within the startup or firm, and it affects the firm’s sales-growth performance. Results suggest [5] that a firm’s management focused on identifying and implementing appropriate strategic processes may be a useful first step in the pursuit of EO effectiveness, i.e., the effective translation of entrepreneurial opportunities into growth.²

A study made on a (representative sample) population of 110 firms [5], has shown that Entrepreneurial Orientation is more positively related to sales growth-rate among firms where major operating/strategic decisions are made in less participative manners (autocratic) and strategies are emergent, rather than planned. It could be anticipated that the extent to which the firm’s strategy emerges over time versus being planned in advance, would impact the firm’s EO-performance relationship due to the incorporation of market feedback and the retention of strategic flexibility. In fact, strategies that emerge and take advantage of feedback from firm operations have been more likely to lead to successful innovations, than strategies that are planned in advance and unaffected by market reactions to the firm’s offerings. Therefore, the study lead to the conclusion that the most successful entrepreneurial firms will be those that incorporate knowledge from their past innovation experiences into their strategies and future innovations (ibid.).

Nevertheless, both the startup Strategic Planning adapted model and Entrepreneurial Orientation still lack a scientific approach that allows startups to efficiently accomplish their objectives without wasting resources, and to progress alongside customer needs. Moreover, to be continuously aware of their progress, allowing them to take “course corrections” towards their goal, whenever necessary and as soon as possible, rather than just “sailing with the wind” and hopping for the best.

1.2. The Emergence of the Agile Approach

In the 1990s the software industry began to understand that the sequential designing process they had been using, from an adapted hardware-oriented model—termed Waterfall and originated in the manufacturing and construction industries—was definitely inadequate. Every study seemed to point towards this inadequacy. A 1995 American study concluded that 46% of the US Defence Department projects did not meet the real needs, although they met the

² For the Strategic Process variables which reflect how firms strategize see appendix 6.2
specifications, and 20% required extensive rework to be usable [17]. Moreover, in a 6,700 projects study (ibid.), 4 out of the 5 key factors leading to project failure were either associated with or aggravated by:

- The traditional Waterfall model;
- Inability to deal with changing requirements;
- Problems related with late integration.

Another study of over 400 waterfall projects reported that 80% of the developed code was actually waste and never used (ibid.). These numbers made clear that the waterfall approach is in fact a risky and expensive approach to build software (and also hardware) systems. Such findings triggered a search for implementing "more agile" alternatives.

Several methodologies, emphasizing close collaboration between the programming team and business experts (promoting face-to-face communication as more efficient than written documentation), self-organizing teams and different ways to craft the code, began then to get increasing public attention. Such methodologies resulted from combinations of old and adapted ideas, and new ideas.

So, later in 2001 was introduced a new paradigm into software engineering which legitimized new methods, practices and tools, new values, new behaviours, new perceptions and perspectives with respect to software development processes. Termed as Agile software development and comprising several iterative and incremental software development methodologies (most popular include Extreme Programming, Scrum, Dynamic Systems Development Method, and Feature-Driven Development), it became a mainstream practice in the software development community.³

³ For more information on the Agile Manifesto formulated on February 2001, which comprehends the Agile methodologies’ common vision and core values, visit http://www.agilemanifesto.org/.
However, lately more and more industries of all types are starting to look like software businesses, as their traditional work process is "infected" by software. Such metamorphose of hardware industries, caused and still is causing them to move along more uncertainty and faster cycle times. Therefore, emerging the need for “more agile” approaches to be applied on hardware development processes.

1.3. Agile Vs. Lean Startup Methodologies

Ten years later, after the core values and principles that all Agile programming methodologies shared had been formulated, those principles got adapted and improved as they became to be found imperfect (or wasteful) and technology evolved. The Lean Startup methodology is then seen by many as the result of such “evolution” and perhaps the successor of Agile—bearing in mind that the Lean Startup methodology also introduces new techniques and a Customer development process, and has its roots in Lean Manufacturing, as seen hereafter.

The major and most relevant alterations that Agile suffered along the years that preceded the emergence of the Lean Startup methodology were the following:

- In Agile it is assumed that in the beginning the Problem is known and the Solution unknown, whereby an iterative process with a continuous feedback is used, in order to really get to the right solution; on the other hand, the Lean Startup assumes that not only the Solution but also the Problem are considered to be unknown. Therefore, a similar iterative
The process used for finding the right solution is used to discover the real and existing problem, which customers wished to see solved.

- In Agile, the pace (progress) is set on increments equivalent to weeks of work and the unit of progress is a line of working code, whereas in Lean Startup those increments are reduced to hours and the unit of progress is customer validation.

- In Agile, it is done Continuous Integration (to continuously integrate the different bunches of code from different authors compile and run automated tests), while in Lean Startup what is done is Continuous Deployment.

- In Agile, tests are made in a “mock production” environment and then developers hope the code acts exactly like it did in the tests once it is released to production (i.e. to the clients); as for the Lean Startup, the product is tested in production with a subset of users (early adopters) and their feedback is gathered, before pushing out the product to everybody (scaling).

- While in Agile the way feedback is sought is through a couple consecutive demos shown to the customers; in Lean Startup distinction is made between what customers say and what they actually do; therefore, although opinions may also be sought, those are empirically validated by actually having visibility of how they used the product, i.e. measuring their behaviours.

- While in Agile “Done” means software that is ready to be deployed and value is delivered on month basis; in Lean Startup it means that the (software) feature has been successfully deployed to production and, most importantly, some learning has been derived from it; value is delivered daily.

Other differences between both approaches are shown in table 1.1 as a matter of curiosity; some of the Lean Startups terms on the table are explained hereafter.

Table 1.1 Other differences between Agile and Lean Startup methodologies

<table>
<thead>
<tr>
<th>Agile</th>
<th>Lean Startup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Roadmap</td>
<td>Lean Canvas</td>
</tr>
<tr>
<td>Product Vision</td>
<td>Product Market Fit</td>
</tr>
<tr>
<td>Release Plan</td>
<td>Minimal Viable Product</td>
</tr>
<tr>
<td>Sprint</td>
<td>Kanban</td>
</tr>
<tr>
<td>Sprint Review</td>
<td>Pivot or Persevere Decision</td>
</tr>
<tr>
<td>Backlog</td>
<td>&quot;To Learn&quot; List</td>
</tr>
</tbody>
</table>
From the developers’ point of view, Agile is an efficient development system as it allows them to stay focused on creating features and technical designs, but it lacks something the Lean Startup approach is very good at: to stay grounded in learning and discovering while constantly iterating toward a plan than works. Attempting to introduce the need to learn into Agile methodologies would ultimately undermine its productivity [2]. Lean manufacturing—concept explained hereafter—faced similar problems when it was introduced in factories, because managers were used to focus on each machine’s utilization rate and factories were designed to keep machines running at full capacity as much of the time as possible. From the perspective of the machine, that is efficient, however from the point of view of the factory’s productivity it is very inefficient at times (ibid.). A proof of this was the outrageous success Toyota had when it applied the Lean manufacturing principles to its production system. Therefore, it is legitimate to say that an attitude grounded in learning and in the product development process, is something only beneficial, which, alongside with the modifications mentioned above and others to be explained hereafter, makes the Lean Startup a legitimate successor of the Agile approach.

1.4. The Lean Startup Approach beyond “Fast Industries”

The Lean Startup approach and its principles have proven very successful in software businesses. Proofs are success cases as, for instance, IMVU and Dropbox companies ([14] and [21]). Moreover, it has also been used by the American Government in several projects, for instance in the healthcare.gov project and in the creation of the new federal agency Consumer Financial Protection Bureau [18]. The question still remaining to be answered in the Lean Startup movement is whether these principles and this approach are completely or partially applicable to slower moving industries, i.e. to hardware product development. This dissertation is exactly an attempt to answer that question.

In summary, the Lean Startup philosophy in software development comprises:

- Selecting only the truly valuable features of a system, prioritizing those selected and delivering them in small batches;
- Emphasizing development workflow, speed and efficiency, and relying on rapid feedback between programmers and customers;
- Using the idea of work-product being "pulled" by customer request, and focusing decision-making authority on individuals;
- Ensuring everyone is productive as much of the time as possible, concentrating on concurrent work and avoiding intra-team workflow dependencies;
And implementing automated unit tests which are written at the same time as the code is written, preventing problems from dissipating.

For a more natural understanding of how much enforceable is the Lean Startup approach to hardware product development, as well as a more adequate transmutation of the method, which may be required, and to perceive some of the concepts and terms it uses, it might help to explore the original roots and meaning of “Lean”. That overview is given in more detail in appendix 6.3.

1.5. The “Lean” Origins

The origins of “lean thinking” go back to a process originated in Japan after the Second World War with Toyota Production System, a revolutionary way of thinking about manufacturing physical goods where the core principle is to avoid and eliminate waste, i.e. eliminate non-value add work—a Lean System.

The word Lean means thin, non-fat, with little waste or no waste at all. In a business context this is translated into speed, efficiency and efficacy, and in the particular context of startups it means the discard of all activities that, by looking at the organization through the eyes of the customer, do not create value.

A Lean System is a revolutionary approach for organizing and managing the supply chain, product development and production operations, which channels the flow of value from concept to launch, and ultimately to the customer’s hands, contrasting with the mass-production system of manufacturing so well-known these days and symbolised by the batch-and-queue methods. It is better approach because it achieves more with less waste, and therefore with less resources. A lean system allows companies to have a clear notion of their definition of value and allows them to line up their value-creating actions in the best sequence possible ([1], pp. 9-98).

Among Lean-Thinking’s principles are: the shrinking of batch sizes, inventory control, acceleration of cycle times, eliminating waste, amplifying learning, delivering as fast as possible, empowering the team and seeing the whole.

Nowadays the challenge for businesses is to discover new approaches which allow reducing costs and improving investment returns while, at the same time, better serving their customers [23]. In order to meet these challenges, organizations have turned to Lean thinking, which seeks the elimination of all forms of waste, continuous improvement, and simplifies business processes. Lean thinking arose from the need to increase productivity in an environment that had few resources. Its implementation proved highly successful, increasing companies’ profits and products’ quality (according to customer demands).

The organizational structure in Lean businesses reflects Lean priorities. Generally, small adaptable teams work together because they eliminate batch and queue wait periods between departments. Additionally, because the team
contains people with diverse skills, they can pool their knowledge and adapt quickly to changing needs. The emphasis on knowledge promotes a culture of learning in Lean businesses, ensuring the workforce can fulfil diverse tasks.
CHAPTER 2. THE LEAN STARTUP APPROACH

2.1. Introduction

The Lean Startup method arises from adapting and applying the lean thinking principles to the context of entrepreneurship and startups, alongside with a foundation on the Agile principles. The motto of Lean Startup is not just about creating a successful entrepreneurial business using different techniques, but about improving virtually everything that is done, ultimately accomplishing and guaranteeing efficiency, sustainability, and success. What distinguishes successful from unsuccessful startups is not necessarily a better initial plan, but the discovery of a plan that works before running out of resources.

The Lean Startup approach can be summarized as a systematic break down of both the business plan and activity into their various parts, which are then empirically tested; it’s a systematic process for iterating from an initial plan—which usually is flawed and never works—to a plan that works, before running out of resources. It builds capital-efficient companies since it allows startups to recognize sooner whether they are making real progress, avoiding waste of resources ([2], pp. 70-79).

First, it is important to define what a startup is according to the Lean Startup methodology. As Eric Ries puts it, a Startup is a human institution with the purpose of creating new products/services, under conditions of extreme uncertainty ([2], pp. 1-15). This means that anyone working in a company of any type or size, under circumstances of very high uncertainty, and whose aim is to create a new business (product or service), is actually initiating a Startup. However, that is not the case if the business being initiated is based on an already existing and similar one, since in this case success is only dependent on execution, which can be modelled ([2], pp. 15-36). It is important to understand that startups’ success is not a consequence of luck, but of performing the rights actions and taking the right steps towards that goal; how and when to make those steps can be learned, after understanding the Lean Startup approach, hereinafter summarized.

The Lean Startup represents a new approach for creating continuous innovation (ibid.). Instead of the usual method which involves a lot of time and effort spent in perfecting a technology followed by an extensive market research—all before releasing the product into the market—this approach applies a learning technique that delivers value to customers and may or not involve the development of an early prototype, i.e. a product first version with problems and missing features. This product first version, or even a mockup (a scaled model with no functionality), is then shipped to customers (early adopters) and charged for. That is called the Minimum Viable Product, MVP. After securing an initial group of customers the product is then constantly changed, updated and improved—as “feedback” from customers, translated into the startup’s metrics changes, is received and assessed—complying with what they really want. That feedback is not based (solely) on opinions and comments on the product from
the customers, but mainly on customer behaviour measurements while they are using the product, so that developers understand how customers in fact use it, and know which features must be improved, removed and how.

Hence, this approach is characterized by a very fast cycle time and a continuous customer involvement; focused on validation, on what customers want but without asking them—based on empirical data.

The core principles of the Lean Startup method are the following:

- **Build-Measure-Learn**: A continuous process of turning ideas into product changes, measuring how customers react and then learn whether to change the product/strategy (to *pivot*) or maintain the current conditions and path (to *persevere*). It begins in the Build stage with a set of ideas or hypotheses that are used to create a minimum viable product. The MVP is then put in front of customers and their response is “measured”. That data is used to derive specific “learning” that serves to validate or refute those hypotheses, which in turn drives the next set of actions;

- **Validated Learning**: a systematic approach to know whether progress is accomplished or not, learning at the same time with every decision made, and making sure all decisions and lessons learned are based on empirical data of customer behaviour;

- **Innovation Accounting**: a quantitative approach that allows understanding if the efforts invested are returning progress, and guides entrepreneurs on setting up milestones and prioritizing work. How can entrepreneurs know that the product-changes they made are related to the results they obtaining? Moreover, how do they know they are extracting the right lessons from those changes? IA helps getting passed these questions. It is a new kind of accounting designed for startups that begins by turning assumptions into a quantitative financial model, ([2], pp. 114-149).

### 2.2. Startups: Why Lean?

Business systems have become overly rigid, failing to take advantage of the adaptability, creativity, and wisdom of individual workers, and there has been an overemphasis on planning, prevention, and procedure, which would only enable organizations to achieve consistent results if we would live in a mostly static world ([2], pp. 275-282). The reason many startups fail seems to be because of taking a solid strategy, based on a solid market research and plan, as a good indicator of likely success ([2], pp. 1-14).

Startups operate under a lot of uncertainty therefore the old management methods are not applicable in their case. The planning and forecasting are only accurate methods when in existence of a static environment, i.e., a long and
stable operating history. Because startups lack a measuring method to assess if they are making progress or, instead developing waste, they often accidentally build something that almost anyone is interested in buying. And if it is waste that is being built, then it does not matter much if startups achieve it on time, nor on budget. Hence, the ultimate goal of a startup is to figure out the right product or service to build and as fast as possible, for which customers will pay.

Instead of making complex plans that are based on a lot of assumptions, it is best to make continuous adjustments through time which is achieved with the Minimum Viable Product and the Build-Measure-Learn feedback loop ([2], pp. 15-25). Contrary to traditional product development, which usually involves a long incubation period and seeks product perfection, the goal with the MVP is rather to begin the process of learning than to end it. The MVP will only contain the necessary features and effort, required to achieve a specific learning, in order to validate one or a couple of assumptions/hypothesis made by the team. However, the team must be sure that the MVP addresses the top problems worth solving (i.e. feasible problems and for which customers will pay) that customers have identified as being important to them.

Through the process of constant-adjusting, it is possible to learn if it is time to make a radical change in the on-going strategy—to pivot—or to maintain it instead, continuing on the current path. Afterwards, the Lean Startup provides a way of scaling and growing the business with maximum acceleration, remaining thoroughly focused on achieving the final goal: build a successful and sustainable business, and fast.

Every startup has a vision and it often is the creation of a thriving and world-changing business. To achieve it, startups employ a strategy, which includes usually: a business model, a product road map, a point of view about partners and competitors, and ideas about who the customer will be. The product is the end result of this strategy.

From the least to the most susceptible of suffering changes are the startup’s vision, strategy and product, respectively. Products constantly need changes—optimizations—until the ideal version is achieved; strategies often reveal to be flawed leading to situations of halt, therefore requiring a course correction in order to test a new fundamental hypothesis—a pivot; however, the vision never or rarely changes [ibid].
In Lean Startup every setback/mistake is taken as an opportunity to learn how to get to the final goal, either by making changes into the product, its marketing or operations or, ultimately, by deciding whether or not to pivot. The modern economy we live in requires failure to exist in a startup [ibid]; a flawed plan and its execution are inherently required to ultimately accomplish perfection, delivering what customers really want. Only by failure it can be learned in the fastest and most economical way where value is (for the customer) and how it can be built into the product [ibid].

2.3. Progress Unit: Learning

In standard management or in any business that we may think of, one is accustomed to measuring their progress by making sure deadlines, quality standards and cost predictions are matched. However, only few of the times does this method represent the actual truth [ibid]. It must be learned which of the strategy elements are truly contributing for achieving the startup’s final goal: a sustainable and prosperous business. Therefore, it has to be understood what customers really want, instead of what we think they should want or what they say they want. In the Lean Startup approach this learning process comprises a rigorous method for tracking down progress, known as “Validated Learning” [ibid], which is able to empirically demonstrate that valuable facts about the startup’s business prospects were discovered.

Validated Learning is a more accurate and faster method than market forecasting or other classical business planning ([2], pp. 37-56). It is about investing efforts that contribute to the essential lessons the team needs to learn, rather than focusing on product improvements which do not contribute to the creation of value—but of waste instead—and only result in the successful execution of a plan that leads nowhere at the expense of valuable resources.

The question that arises is: Which of the team’s efforts create value and which are wasteful? From the Lean thinking perspective, value is defined as whatever
provides benefit to the customer; whereas waste is anything else other than that. However, in any startup, what the customer might find valuable is not known initially—hence being a startup, where uncertainty abounds—this being the reason why validated learning is so vital: based on empirical data from real customers, with validated learning one is able to demonstrate the startup’s progress. Progress is reflected in the startup’s metrics in a positive, null or negative way, reflecting the outcome of undertaken product or strategy changes, and making thereby clear whether such changes are desired by the customers.

Wherefore, **learning is the progress-unit of a lean startup.** By learning what customers want and do not want (validated by the effects in the metrics), one is making actual progress and getting closer to a valuable and desired product, and a thriving business. In this sense, all the effort that is not contributing to the learning of what is valuable, from the customer point of view, can and must be eliminated.

The correct way of evaluating a startup’s productivity, in accordance to the Lean Startup, is not on how many things are being built but on how much validating learning is accomplished. It is then clear that in a startup, to finding out which efforts are not wasteful and what the customer might find valuable, a **Scientific Approach** is required, and that is exactly what the Lean Startup approach provides.

### 2.4. Becoming a Lean Startup

A startup following and applying the concepts previously explained (fast cycle: B-M-L loop; MVP; validated learning; etc.), will certainly struggle with many first questions ranging from “Which product features to give priority?” or “Which of them are essential for the success of the product?”, to the uncertainty of which customer opinions to listen to and value. The MVP technique gives room to such uncertainties; their answers reside in a trial-and-error solution, with **progressive learning** [2].

Instead of diving into a slow process of market researching and business plan developing, by building a first-version product or even only a mockup or other technique hereinafter mentioned, the startup’s team is able to get accurate and real data, much faster and efficiently about customer demands, needs, and unexpected behaviours that could not have been predicted in any possible research (ibid.). Moreover, this “proximity” between the startup and the customer, in such early stages, allows for a fast interaction between both, resulting in a faster and sooner deskewing—read as skew correction—of the startup’s running-direction towards its goal. With this methodology, the startup’s assumptions may then be easily put to test and thereby validated or rejected **fast**.

There are 2 vital assumptions that any entrepreneur has to make in a startup:
• The value hypothesis;

• The growth hypothesis.

The first one defines what customers see as valuable for them; the second establishes the dynamics involved in the product/service spreading-process inside the target market (which are mentioned hereafter in this work). Both of them cannot be tested in a more efficient and accurate way than with a MVP and empirical testing with real customers, as explained earlier (ibid.). If the results of such tests are not in accordance with expectations or requirements, it means something is wrong with the current deployed strategy; therefore fast qualitative customer “feedback” is required in order to proceed with inevitable and much needed strategy changes.

Even when changes produce worst scenarios translated into undesired metric changes, ulteriorly, they are also instructive and add something to the startup learning process. This is what I was referring to when I earlier referred to the advantage of the startup-customer “proximity”, as being the possibility of carrying out a “direction deskew”—a fast and confident strategy change pointing the startup to a new direction, closer to the ideal.

All the process of creating a product is viewed as an experiment. First, the elements of the plan which are merely assumptions are identified; then these assumptions are tested through minimum viable products. Meanwhile the startup is already running, although on a micro-scale with a special type of first customers (early adopters) but long before an official plan would ever have been set. Moreover, the startup is able to collect valuable data from real customers, allowing it to engage in a continuous product improvement process, adding more and better solutions each time, through each iteration of the B-M-L loop. With this approach, a product can be built in a very short time with considerably fewer (financial) resources, and also by taking advantage of some of the many helpful tools\(^4\) available nowadays.

In the product development of Lean Startup, it is taken into account what has been stated until here but it is also given great importance to determining what is value-creating and what is value-destroying. Instead of being based solely on what product manager and engineers want and believe it is the right thing to build, the product development follows 4 simple questions (three more before jumping to the only question that the standard product development puts):

1. Is the problem, for which the product is the solution, recognized by consumers?

2. Would consumers buy the solution if there would be one?

3. Would consumers buy the solution from us? Why?

4. Is the team able to develop such solution?

Another source of waste that still is created during the first stage of a startup in the current days is, in fact, the elaboration of a business plan the way we know it: several weeks or months invested in writing a 60-or-more-pages document, based on untested hypotheses. Since most initial plans are likely to be proven wrong, it only makes sense that a less static and rigid plan should be used. The one used in a Lean startup is a one-page business model diagram termed as *Lean Canvas*; it helps deconstructing the business model into nine subparts which are then systematically tested, from the subpart posing the highest risk to that with the lowest risk. The advantages of such plan format (see Figure 2.3) are: reduced time spent on creating a business model, which allows to outline multiple business models in very short time period; the concise nature of it obliges entrepreneurs to discovering the real essence of the product they want to sell; the unquestionable portability which in turn ultimately makes the business plan to be easily shared, meaning it will be read and evaluated by more players, and probably more frequently updated and easily improved.

---

5 See Appendix 6.5 for an overview of how to prioritise risk

**Fig. 2.2 Three stages of a startup (adapted from [3])**
Fig. 2.3 The one-page business model “Lean Canvas” (adapted from [3])

2.4.1. Making Progress

The “skeleton” supporting the lean startup model is the Build-Measure-Learn loop. The core objective is to minimize the total time through the feedback loop, only by achieving this is one able to truly take advantage of the model. As has been stated previously, the growth and value hypothesis are done by every entrepreneur and these are two of the most important leap-of-faith assumptions on which the whole startup’s strategy relies, therefore requiring testing as quickly as possible—the startup’s strategy is always based on assumptions.

To test the leaps-of-faith, one must go through one loop cycle, entering in the Build phase with a MVP, which will allow a full turn of the B-M-L loop with the minimum amount of resources (e.g. time, financial resources) and helping entrepreneurs begin the learning process as quickly as possible. Unlike a prototype, the MVP is design to answer questions not only of technical and design natures, but also questions regarding the fundamental business hypothesis, measuring its impact on customers through several different and suitable metrics\(^6\) (depending on the business as we will see hereafter). When

\(^6\) See Appendix 6.4 for an example of very popular metrics in software businesses (AARRR metrics)
entering the Measuring phase, the important thing to know is whether the efforts invested in the product development (Build phase) are resulting in real progress. Once the assumptions that the MVP was supposed to test have been confirmed or refuted, progress has been achieved, and either a new minimum viable product is built to test other assumptions or the solution space has to be changed. If many assumptions turned out to be refuted, then probably the issue is the wrong problem space, which will need to be changed (pivot).

![Diagram of problem and solution spaces](source: own)

**Fig. 2.4** Solution and Problem spaces; source: own

### 2.4.2. “First-version” Products: MVPs

Before a new product is sold to the mass market it needs to be sold to a first share of customers of the product’s adoption lifecycle: the early adopters (see Figure 2.5). These represent an ever present special type of customers, who accept a product to be an 80% solution, as long as they are the first ones to get hands on the product’s first version either to show off or to gain a competitive advantage by risking acquiring something competitors do not have yet ([2], pp. 92-114). This means that if the product has additional features beyond the considered basic ones—demanded by early adopters—the team is then building a form of waste and, at the same time, risking losing these group of customers, since there might be no thrill left in buying early a product that is suitable for everyone to adopt. Also an advantage of this approach is that early adopters, using their imagination, suggest and fill in the product’s missing features, which is obviously valuable and most advantageous.

“In a great market, a market with lots of real potential customers, the market pulls the product out of the startup.”

—Marc Andreessen
Building products today is not as physical and resource intensive as it used to be. With the emergence of the Internet, open source software, cloud computing, and globalization, we need fewer resources than ever to get a product to market ([3], pp. XXI-XXVIII). A minimum viable product, as long as it delivers value to the customer, may range from a simple smoke test—a rudimentary version preview of the product only comprising very basic features, e.g. an advertisement or video—to an early prototype with “bugs” and missing features. How to decide to which extent of complexity shall the product development team build the MVP, requires judgement, but most of the times there is the tendency to overestimate of how many features are needed (ibid.). Any feature or effort that does not directly contribute to the learning sought should be removed, that means eliminating “nice-to-haves” and “don’t-needs”, but making sure enough value is delivered to justify charging for it—the product unique value proposition must be delivered.
An example of a MVP technique, termed “concierge MVP” (ibid.), is the startup called “Food on the Table” which, based on a website user-interaction, envisioned the creation of weekly meal plans and grocery lists according to the client’s preferences, and to the locally-available grocery stores near to them. Clearly a very elaborated service when we think about the required work involved in order to achieve it: databases’ maintenance of practically every grocery store in the country accounting with weekly sales on each one; also, groceries had to be matched to appropriate recipes and then appropriately customized, tagged, and sorted, etc. What the CEO of Food and Table startup and his team did was to look for a willing first-customer—an early adopter—and give him/her dedicated individual treatment proceeding as if the customer was actually taking advantage of the fully functioning system, i.e. the website cross-checking all the groceries’ databases updated information, the delivering service, etc. Instead of marketing themselves to millions, they sold themselves to one. Although their efforts did not seem to be leading to anywhere tangible (as they had no product, no meaningful revenue or databases of recipes), viewed from the Lean Startup perspective they were making huge progress. That is because each week they were learning more about what was required to make their product a success, which allowed them to have more customers. Soon the overhead of serving customers one-on-one started to increase becoming unfeasible to acquire additional customers, so the CEO and his team start to invest in automation: delivering the recipes and shopping list via e-mail instead of via an in-home visit, starting to parse lists of what was on sale automatically via software instead of by hand and taking credit card payments online instead of a handwritten check.

The bottom line is that the team was able to test the leap-of-faith assumptions of the startup’s growth model and focus on scaling something that was working, rather than trying to invent something that might work in the future; ultimately building a sustainable and successful business. Instead of a product or its simple version, the “concierge MVP” provides entrepreneurs with a valuable learning activity for testing the business hypothesis.

Another very useful technique is the split-testing. A split-test experiment consists in offering to distinct group of customers different versions of a product (for instance, with distinct features or lacking some) at the same time and observing the changes in behaviour between the two groups which helps teams refine their understanding of what customers want and do not want.

Other MVP techniques are:

- **Smoke tests** - an advertisement or sign-up form for a product that does not yet exist, it can be used to gauge interest and market size;

- **Release 1.0** - an early prototype, with just enough features in order to test the most important business assumptions;

- **“Wizard of Oz”** - where a user believes an action is being performed by a computer, when in fact it is being performed by a human;
• Mock-ups - a scale or full-size model, a non-functional prototype;

• Simulations

2.4.3. Measuring Progress: Innovation Accounting and Cohort Analysis

As said previously, if the startup is successfully building something of no interest to anyone, then it is worthless if it does it on time and on budget. In order to measure progress, the Lean Startup model takes advantage of the innovating accounting to create learning milestones (learn phase), useful for entrepreneurs to assess their progress objectively and accurately.

Although the execution of the plan through the Build-Measure-Learn loop follows that order, i.e. first ‘Build’, then ‘Measure’ and then ‘Learn’, the planning necessary for engaging the B-M-L loop has the reversed order. First it is necessary to know what must be learned; then knowing which metrics to measure (innovation accounting), one is able to see whether validated learning (efforts translated in relevant-metrics changes) is achieved or not; and only then it is possible to determine what product must be built so that the experiment is executed and the intended measurement is accomplished.

![The Build-Measure-Learn feedback loop](Image)

Fig. 2.7 The Build-Measure-Learn feedback loop

Experimenting is critical for Startups. Risks are tackled through experiments; however one can never completely eliminate them through a single experiment. Before product/market fit, qualitative learning is crucial ([3] pp. 58-68). A startup needs to figure out which techniques will work in its unique circumstances and
determine the right questions “to ask itself;” it needs to get extensive contact with customers in order to understand them and determine which assumptions require the most urgent testing. The goal is not to acquire a definitive answer but instead, it is to certify that the team understands the potential customer (i.e. to have a clear knowledge on the customer’s problem or problems, for which they would pay for a solution), therefore being able to draft a customer archetype, which will be critical to guide the product development team. The customer archetype is a provisional profile of the proposed target customer, a hypothesis, which allows certifying that the team’s daily decisions’ priorities are in accordance with the type of customer they wish to satisfy, and it remains an hypothesis until validated learning has shown the team is able to serve that type of customer sustainably. There are many techniques for building accurate customers archetypes ([2], pp. 79-92), such as design thinking, interaction design, rapid prototyping and in-person customer observations.

During the learning process of going through several quick cycles of the Build-Measure-Learn loop, some of the MVPs built will result in positive changes in the startup’s metrics—proving right the assumptions tested by that MVP—but others will originate negative results. In order to prove there is not misguidance and to know whether entrepreneurs are making a better product and accomplishing validated learning, the Lean Startup model applies innovation accounting. It is a different accounting model which allows entrepreneurs to understand the causes and the dynamics behind the metric’s changes. It comprises measuring everything as a cohort (hereafter explained), focusing on a single metric but at the same time monitoring and relating the full customer lifecycle, and building a feedback loop with customers to rapidly generate hypothesis for the causes of the changes obtained on the cohort analysis of that metric. In other words, contrary to traditional/standard accounting, innovation accounting allows to understand the reason/work/lessons-learned behind the numbers; not only the gross numbers (e.g. revenue, total number of paying customers, total number of orders and all the progress, etc.), but rather all the process and path covered that justify those numbers.

Every product, even when unsuccessful, normally has some traction, i.e. interested and loyal customers (ibid.). This is one important reason why entrepreneurs must know what exactly to measure when trying to understand if they are in the right path, otherwise they risk getting stuck in “treacherous soils” ultimately failing the startup's goal.

Innovation accounting consists basically in 3 learning milestones:

1. Finding out the startup’s current status by building MVPs and getting real data (e.g. conversion rates, sign-up and trial rates, customer lifetime value, etc.) measuring how customers behave—defining a baseline;

2. After a baseline has been established, follows targeting every effort of the startup’s initiatives (i.e. product development, marketing, etc.) at improving the growth driver of its growth model (which primarily may be, depending on each case: the profitability of each customer, the cost of acquiring new customers or the repeat purchase rate of existing
customers) in order to find out if it is possible to **improve the metrics towards the ideal**, through product optimizations, and making use of the B-M-L loop. To demonstrate validated learning, positive changes must be reflected on that growth driver;

3. Whether the attempt of moving the startup’s baseline towards the ideal was successful will be translated, respectively, in the team **maintaining or changing (pivoting) the strategy** at hand. When a company pivots, it starts the process all over again, re-establishing a new baseline and then tuning the growth engine from there.

In order to be able to use innovation accounting and guiding the startup’s actions, one needs need to know the right metrics to rely on. Startups must track the metrics that are critical to their growth strategy, and avoid focusing on the top-line numbers (e.g. total number of customers, total number of downloads or increase in overall revenue) that allure entrepreneurs to think they are making good progress—these are the “vanity metrics” [2].

![Cumulative metrics](image)

**Fig. 2.8** The traditional hockey stick graph: Vanity metrics [2]

The traditional funnel reports also are not the right choice to get the numbers. They consist on specifying a period over which an event occurs and count the number of occurrences. They are better suited at tracking short lifecycle events (measured typically in minutes, like acquisition or activation) than long lifecycle
events which are measured in days or months (for instance, revenue), and are inappropriate when the intervals between events fall outside the reporting period. Other problem of the funnel reports is that with them it is hard or even not possible to relate results to actions taken in the past, and they do not allow measuring a split test ([3], pp. 121-127).

![Funnel Report](image)

**Fig. 2.9** Typical funnel report on a month period for a downloadable product [3]

One of the most important tools of startup analytics is the **Cohort Analysis**. This technique applied to the startup’s metrics analysis consists on dividing metrics in relevant groups (cohorts) based on any property attributable to a user (e.g. operating system, logged in, registered but did not log in, etc.). Also for the case of split-testing, customers are divided into cohorts, each of them being in contact with products that differ only in a design detail or feature. A cohort analysis allows entrepreneurs to have a much clear and quantitative view of the business, to quickly test product optimizations/assumptions (for the case of split-testing), and to have some predictive capacity since the cohort-based reports show clear trends and cause-effect relations.

Cohort-based reports are at the core of learning metrics, they turn complex actions into **people-based reports** ([2], pp. 114-149). Each cohort analysis basically tells teams, among the people who used their product in that period, how many of them exhibited each of the behaviours they are interested in. **The Lean Startup couples the funnel analysis with the cohort analysis.**

![Weekly Funnel-Cohort](image)

**Fig. 2.10** Weekly funnel-cohort by join date [3]
In the Lean Startup methodology, metrics to be appropriate need to be (ibid.):

- **Actionable**: they allow teams to draw clear cause-and-effect relations between actions taken and metrics changes;

- **Accessible**: reports must be simple, use tangible concrete units so that everyone understands them (time is not wasted trying to understand them), and be available and shared with everyone in the startup so that teams experience a flow experience—which is important, as explained hereafter in appendix 6.3;

- **Auditable**: data must be credible to employees, leaving no room for doubts about its veracity and allowing to easily test its consistency with reality.
2.4.4. **Maintaining or Changing Strategy**

When startups begin to run low on financial resources, they can extend their attempt of achieving their goal—to build a sustainable and profitable business—by either cutting costs or raising additional funds. In Lean Startup to know how much time a startup has left until it either achieves “lift-off” or failure is to know how much pivots it can still perform.

A pivot is a change in the strategy—usually a change of problem space—designed to test a new fundamental hypothesis about the product, business model, or growth strategy (engine of growth). It is at the core of the Lean Startup methodology, happening constantly in any growing business and even in already successful businesses. Not to forget that the Lean Startup assumes both the solution and the problem as unknowns, therefore several pivots are expected to occur before finding the right problem worth solving.

There are different types of pivots depending on the situation:

- **Zoom-in Pivot:** what previously was just one feature of the product, becomes the main or the product’s only feature, i.e. becoming the product itself;

- **Zoom-out Pivot:** the opposite of the zoom-in pivot. What was expected to be the main functionality or feature becomes just one of many, of a larger product;

- **Customer Segment Pivot:** the product solves a real problem but of a type/group of customers different than initially planned;

- **Customer Need Pivot:** the case when the target customer has a problem worth solving and which the team would be able to solve, however it is not the one originally planned by them;

- **Platform Pivot:** refers to a change from a product, initially, thought as an application to a platform, or vice-versa;

- **Business Architecture Pivot:** consists on a change from a high margin and low volume business (business to business) to a low margin and high volume one (consumer products);

- **Value Capture Pivot:** changes in the revenue model, i.e. in the way the startup captures value, which may lead to further changes in the business, product, and marketing strategies;

- **Engine of Growth Pivot:** consists in changing the startup’s growth strategy in order to achieve a more profitable situation and faster growth;
• **Channel Pivot**: consists in changing the sales channel, for instance to customer direct sale, with effects on the product’s price, features and competitiveness;

• **Technology Pivot**: more common in established companies, this pivot happens when teams discover they are able to achieve the same solution with a different and more favourable.

### 2.4.5. Small Batches and Continuous Deployment

Until lately, many companies were disciples of the scientific management movement initiated by Frederick Winslow Taylor in 1911 which brought us to the prosperous times we live today—despite the current economic crises ([2], pp. 272-285). This way of managing is based basically on: improving the individual workers’ efficiency, relying on unexpected results (good or bad), organizing work into tasks, and compensating workers on a task-plus-bonus basis—which is inefficient regarding its motivational purpose, according to Csíkszentmihályi research’s conclusions, overviewed in appendix 6.3. By focusing on functional efficiency, workers lose sight of what really matters: to learn what is still unknown and improve each time on the method used to achieve that goal. However, Taylor has also foreseen that companies should and would be managed in future—the present days—at the system level, focusing on the system’s efficiency rather than on the individual’s.

In the Lean Startup methodology, by reducing the batch size to a single-piece, startups are able to tackle quality problems much sooner, minimizing the waste of resources, and obtaining validated-learning much faster, ultimately learning quicker how to build a sustainable business.

The standard “waterfall” development methodology that product development teams have used for years ([2], pp. 224-253) is represented in Figure 2.12. It is a linear, large-batch system that relies on proper forecasting and planning for success, inappropriate for today’s rapidly changing business environments. Looking at Figure 2.12 becomes clear how wrong it is that most of the learning happens in the final stage of the product development process. The learning that is being acquired before release is also valuable, although quantitatively small, but is surely not about the customers ([3], pp. 111-120). The Lean Startup does not eliminate the stages of Development and Quality Assurance, but rather shortens the cycle time from requirements to release.
Although it may seem counterintuitive to work by the unit and not by stages with a big group of several units, it is a wrong idea since it is not being taken into account the extra time spent with the additional work: moving around the large piles of half complete products (whether they are apparatus or lines of code of many features); sorting and stacking; undoing/repeating a certain action in the whole batch, because only by the end of finishing step ‘A’ in the whole batch did the worker realise that there was a problem when moving from that stage to the next stage ‘B’ (quality problem identified too late for preventing waste), etc. Moreover, the large-batch approach does not enable prevention against the enormous waste created when customers decide that, for some reason, they do not want the product ([2], pp. 184-206).

Repeating the same task over and over with a large batch of units, progressing from stage to stage, will indeed result in higher individual performance, however, in a startup or any successful business this is not as important as the overall system’s performance (ibid). Process-oriented work is the one that must be fostered in order to improve the process by which the startup accomplishes its goals (the Lean principle of ‘Perfection’) rather than just achieving a set of
goals that are a result of immediate necessity and do not contribute for more than just that.

Just as with the Toyota Production System, in the Lean Startup context engineers and designers work side by side on one feature at a time, testing it once it is ready and releasing a new product version to a small group of customers for immediate assessment with customer validation. This allows teams to realise sooner about defects preventing bigger problems later. This use of small work-batches enables another technique which speeds up the cycle time from requirements to release: Continuous Deployment. It consists on releasing pieces of code or improvements continuously throughout the day. This way, time wasted while waiting for code to pass from one stage to another (coding, testing and deploying stages) is avoided. Undeployed code means more integration, more coordination and more planning, reducing the ability to react quickly ([3] pp. 111-120)—the same way that in lean manufacturing time is not wasted in transporting products from one place to another since there is a constant flow. As soon as changes are ready they are on the way to the production environment, moreover an immune system that performs automated tests and real time alerting is implemented (see Figure 2.14). Whenever a problem is found the team stops deploying and gets involved in diagnosing and fixing the problem.

![Continuous Deployment Principles](image)

Fig. 2.13 Continuous Deployment Principles; source: own

This technique has been used in several companies, for instance IMVU, Flickr and Wealthfront (ibid.). It is relevant to point out that although this technique deploys code into production in small batches, the code does not have to be live for the users (software release is not equal to marketing release).

Continuous Deployment also comprises an “immune system” which in the presence of a defect and through a set of special tools and techniques (e.g. continuous integration server: BuildBot; running tests locally; and real-time alerting), performs the following:

1. The defective change is immediately and automatically removed;
2. Everyone on the relevant team is notified of the problem;
3. The team is blocked from introducing any further changes, until the root cause of the problem is found and fixed (through the Five Whys method).

- Code in smaller batches
- Server monitoring
- Real-time alerting
- Five whys
- Testing is everyone's responsibility
- Continuous integration server (auto-test)
- Use customer lifecycle to prioritize tasks
- Deploy scripts that allow pushing code to the production server and roll it back to the last release

**Fig. 2.14 Continuous deployment cycle [3]**

The Pull approach of lean manufacturing applied to startups translates into reducing their Work-In-Progress inventory which in this context (startups) is more intangible than in an established business. In a Lean Startup context the WIP comprises all the work put into designing a MVP until it is shipped to customers for validation (e.g. assumptions yet to validate, any kind of planning, incomplete designs, etc.). The product development process is nothing more than a response to pull requests from the team’s *hypothesised customer*, in the form or experiments that need to be run. In Lean Startup no product or service should be produced until a customer “asks” for it.

### 2.4.6. Customer Development

A particular characteristic of Lean Startup methodology is that instead of having department labels, for instance “Engineering,” “Quality Assurance,” and “Marketing”, the organization is made in 2 cross functional teams, the *Problem team* and the *Solution team*. Each employee is given 100% responsibility for their own time, leading to a culture of learning and creating transparency. If someone knows something is not going to work out, then it is 100% their responsibility to speak up and prevent it from happening. Everyone’s job description is the same: to use their best judgement to do what is in the best interest of the company.
Instead of the Sales, Marketing and Business teams, there is the Problem team, in charge of Customer Development and mostly involved with “outside-the-building” activities such as interviewing customers.

Instead of Engineering Operations and Quality Assurance teams, there is the Solution team that, for example, is responsible for defining the business model and creating MVPs.

Customer Development is a continuous feedback loop with customers, parallel to the product development cycle, which takes care of:

- Finding out who the customer is;
- Successfully engaging customers so that teams acquire the answers that will validate the business uncertainties and assumptions;
- Find out the problem for which customers would pay for a solution;
- Identify a MVP that resonates with customers and find out whether it is scalable;
- Test and validate elements of the vision.

![Customer Development diagram](image)

**Fig. 2.15** Customer Development diagram [3]

The concise description of each Customer Development step from Figure 2.15 is the following:

- Discovery: Hypotheses testing, namely problem and product concepts; understanding of the problem, and making sure that the product answers the customers’ needs (Do customers want the product?);
- Validation: Validation of sales and distribution processes; first revenue by selling the early product (Do customers pay for the product?);
• Creation: The creation of end-user demand and filling the sales pipeline; marketing launch;

• Building: Build a repeatable and scalable sales process; scale via relentless execution (transition from a learning focus to an execution focus).

2.5. Boosting the Engine of Growth

Finally, once the right problem to solve has been discovered and product adjustments are becoming common, the focus shall turn to boosting the business growth, namely by knowing which metrics to focus on the efforts. There are 3 main engines of growth, i.e. growth strategies:

- Sticky growth models: high retention focus;
- Paid growth models: high margins focus;
- Viral growth models: referral focus.

Simply put, the first one is based on the fact that once customers start using the product they continue on doing so, as for instance the Mobile Network Operators. Companies using this strategy must focus on having low churn rates (i.e. the amount of customers that fail to remain engaged with the product in a certain time frame). If customer acquisition rate is higher than churn’s then the company will grow, therefore focus must be given to improving customer retention, for instance by giving customers incentive to check back often on the product ([2], pp. 206-224).

On the other hand, on paid growth models customers are willing to pay more for a product than it costs to reach them with that product. That amount of money they pay over their lifetime is termed customer lifetime value, LTV, which can be invested in advertisement to reach more potential customers. Therefore the margin between the cost per acquisition, CPA (i.e. the amount invested in ads divided by the number of acquire customers), and the LTV determines how quickly this type of engine of growth turns, and focus must be given to the differentiated ability of monetizing customers, since because of market competition every source of customer acquisition will tend to have a high CPA (ibid.).

The viral engine of growth is that where product awareness spreads rapidly and unwillingly from customer to customer, similar to what happens with viruses, as a side effect of using the product. The speed of this type of growth strategy is determined by how many new customers will use a product as a consequence of each new customer who sings up—the “viral coefficient”. The higher the viral coefficient is, the faster the product will spread; as long as it is higher than 1.0 the company will grow exponentially. In this case, focus must be given on
increasing the viral coefficient more than anything, as slight changes on it will translate into dramatic changes (ibid.).

The best way of deciding for which engine of growth is to understand how customers behave with the product (MVP), trying to identify particular usage patterns (implicit virality, recurring use or one-time-use behaviours) ([3] pp. 155-168). It is important to understand that other focus given to other aspects, other than those referred in the previous paragraph for each type of engine of growth, is most likely a form of waste for a lean startup strategy. For instance, monetary exchange in the viral engine of growth does not drive new growth but only proves that customers value the product enough to pay for it (which if the viral coefficient would be higher than 1.0 then it was already known), and most likely will slow down the “viral” spreading of the product.

2.6. Summary of Steps to Take

The essence following a Lean Startup approach when creating a business can be summarized in following steps:

1. Decide on the type of growth strategy:
   - Viral, Paid or Sticky;

2. Document the initial plan or plans (business model) using the Lean Canvas;

3. Identify which hypotheses to test first: the riskiest elements of the startup’s plan, i.e. the leap-of-faith assumptions (the parts on which everything depends: value hypothesis and growth hypothesis). These give rise to tuning variables that control a startup’s engine of growth;

4. In parallel, develop a process to manage Customer Development;

   5.1. Once clear on the leap-of-faith assumptions, use innovation accounting to objectively prove that the business will become sustainable and to test those hypothesis:

   5.2. Enter the Build phase of the Build-Measure-Learn loop as quickly as possible with a minimum viable product (MVP), and establish baseline metrics;

   5.3. Target every effort at improving the startup’s growth driver, in order to find out if it is possible to improve the metrics from the baseline towards the ideal, through product optimizations;

   5.4. (Measure and Learn phases) If customer insights are being translated into tangible metrics changes due to product improvements/changes, and those metrics are moving towards the ideal (i.e. leap-of-faith assumptions proven right), then the team shall maintain the strategy (problem space), eventually performing small adjustments (solution space); if there is no metric changes
then the strategic (leap-of-faith) hypothesis is considered to be proven wrong and the team must change the strategy at hand, reformulate the plan, (pivoting) and re-start the process.

6. After proving the unit economics of the business in micro-scale, scale up.

**Fig. 2.16** Steps summarizing the Lean Startup approach; source: own

**Fig. 2.17** Workflow to follow when building a Lean Startup (adapted from [3])
CHAPTER 3. INTERVIEWS

3.1. Questions

The interviews consisted on a group of 15 questions put to the entrepreneurs preceded by a 15 minutes PowerPoint presentation overviewing the main aspects of the Lean Startup methodology (scientific approach). The 15 questions worked as guidance to the interview, which in each case progressed differently until the following questions were covered:

Question 1:
The Lean Startup provides a scientific approach to creating and managing startups, and a method for discovering and building the right products, aiming to eliminating all forms of waste. In your opinion, what constraints do you immediately think of, if you were to apply this method to your business?

Question 2:
Minimum Viable Products (MVPs) are simple/incomplete/mirror product versions that offer the value and solution the customer seeks, and aim to prove or disprove assumptions on product features or optimizations. A very first MVP can be, for example, a photo with the Value Proposition explained or implicit. How would you build your MVPs in your business? (Prototype, mockup, smoke test, “Wizard of Oz”, or other) Why?

Question 3:
In your opinion do you think it is possible to have fast iterations cycles (Build-Measure-Learn loops) with hardware development? What about in your business in particular?

Question 4:
Continuous deployment consists in constantly improving products and releasing them to production as defects or space for improvements is discovered. Rapid prototyping is one solution for applying continuous deployment in slower-moving industries, for example: 3D computer-aided modelling, moulds injection and shifting the hardware product’s value to software. In your opinion, do you think any of these techniques could have been used in your business?

Question 5:
In order to have fast iteration cycles and continuous deployment, small batches of work are required. Please give an example of how big is the batch size of work in your company. Do you think it is possible to have small work-batches in hardware development?

Question 6:
What happens if there is a problem with the product when the customer is using it? How are you able to detect it and correct it?
Question 7:
How do you think you could deploy real-time problem-monitoring and problem-correction in your products?

Question 8:
How is or was the process of validating the startup’s assumptions regarding the product (“Is the product offering the right solution to the right problem that our customer has?”) and the customer (“Are we sure who our customer really is?”)?

Question 9:
In your opinion, regarding your business, do you think you could have validated those assumptions with empirical results from real customers? Could you have had customers who would try first-version products and then, from there, you would iterate and improve those first versions?

Question 10:
Which are/were the metrics used to assess the company’s progress when releasing new products or product improvements?

Question 11:
In software products, many customer actions can be “monitored” and therefore a clearer understanding of the customer’s opinion and product-usage can be obtained. In your opinion, how could that be adapted to hardware products?

Question 12:
In your opinion would split-testing be or have been a viable technique for testing product improvements with your customers in your company?

Question 13:
Are product optimizations common in your products? How are they processed from first deciding which optimizations to make to finally have them applied in the product?

Question 14:
Which steps must be taken from a new product idea to finally be available to all customers, or rejected?

Question 15:
Is there any relevant question that should have been made or any uncertainty that you might have about applying this method to your case?

3.2. Interview Results

The report of each interview is organized in 3 sections: the first giving a panorama of the company’s case; the second giving the interviewee’s point of view regarding the applicability of the Lean Startup approach to their startups; and the third section providing the interviewee’s view on whether the 3 main
techniques (MVP, Continuous Deployment and Split-testing) used in this approach and whose applicability in hardware development is, indeed, questionable, could have been used and adapted to their businesses.

3.2.1. Alteraid Interview

Interview made to Professor Jesus Alcober, Audiovisual Director of I2CAT Foundation, Associate Professor of Universitat Politècnica de Catalunya and Co-Founder of Alteraid Company.

Alteraid SL is a spin-off from Universitat Politecnica de Catalunya (UPC) that has developed aaaida social network and sensors, providing a unique and innovative way of ensuring peace of mind to families, care workers and clinicians who care for elderly people living alone. The company offers a service where elderly are unobtrusively monitored by special dedicated sensors which are connected to a social network of family members, clinicians, etc., giving them the peace of mind that all is well with their elderly relative.

aaaida provide a set of solutions and tools to manage the quality of life of their customers’ relatives, through a kit of sensors that monitor unobtrusively the daily life of those people and notify the customer when it is out of what you consider normal. Most sensors can be easily obtained as final solutions that incorporate the wireless capacity in order to be directly deployed in a wireless sensor network. Alteraid aaaida is a real use case of the sensor network in healthcare.

Website: http://eng.alteraid.com/

Address: C Esteve Terradas 1, PMT-Parc UPC 08860 Castelldefels, Barcelona (Spain)

3.2.1.1. About the company’s case

Regarding the validation of Alteraid’s assumptions about the customer and the product, as a first approach, they performed surveys on a group of persons in order to validate the worthiness of the problem: whether customers wanted and were willing to pay for the solution. This group’s profile and family conditions matched the characteristics of the company’s target customer. The second approach was to look for specific persons, in a hospital, who had specific health problems and to whom the product could offer a solution and actually be tested in all its extent. By getting “out of the building” they got valuable feedback regarding not only the web platform, to which the sensors sent the data to, but also about which sensors could be developed and how they could be improved. That feedback was achieved by assessing and listening to each patient’s
medical needs. Finally, Alteraid developed and built sensors and tested them with those customers.

The steps taken by Alteraid in the presence of a new idea to be tested were the following: implement it in a prototype, then test in laboratory, and then test in a couple of control-homes. The time it took to conclude this process was significant: “Uff! Depending on the staff devoted to development of the prototype, it will take at least three months.” After this testing phase, according to the test subject’s feedback and the development team’s assessment, a decision was made regarding the scaling of the new improvement (possibly with further improvements) to all customers. Instead of genuine early adopters, the group of subjects with who Alteraid did its testing were actually chosen and searched for: persons with specific chronic diseases and clearly benefiting from the product. “We are quite involved in the health care world so it is easy for us to get close contact with people who suffer from chronic diseases.”

In terms of how common product optimizations were in Alteraid, naturally they were not as common as it occurs in a software product. However, Alteraid had already performed some product optimizations, but more in an initial phase. For example, they first implemented sensors with fractal antennas, afterwards discovering that it would be cheaper and still leading to the same desired results if they had substituted the fractal antenna with a much simpler one.

In case there was a problem with the product, and if it only affected its performance and not preventing the product from performing its function, no action would be taken as they were interested in the sensors to be cost-effective. However, if it actually prevented the product from achieving its purpose, then they would send a technician to the site in order to fix the problem or replace the product with a new one.

Alteraid was currently not able to detect and monitor in real time possible problems with their products. Professor Jesus had the opinion, however, that it could be possible by having sensors with more processing capacity, but the cost involved would be high, probably making it unfeasible. The only way Alteraid had for detecting problems was to get customer feedback, i.e. a complaint for instance.

3.2.1.2. Applicability of the Lean Startup approach

Regarding the applicability of the Lean Startup to the Alteraid case and a physical products business in general, in the opinion of Prof. Jesus the main problem was of financial nature. “Which is the cost of this process compared with the revenues of this MVP that we are building?” According to him, in a bootstrapping strategy the initial revenue and costs are critical aspects to have into account. Although the method seemed very attractive to him—since testing is accomplished at the same time that feedback is acquired—he was of the opinion that to have fast iteration cycles and continuous deployment would probably be something difficult to achieve in hardware product development. At
least in the same manner as it is achieved in software development, it would be most likely impossible to achieve.

For Prof. Jesus the issue was also in the need for more initial capital than in the case of a software product development scenario. “**Convincing investors would be an issue as there is not a clear plan at first.**” More capital is required up front, whereas in the classical view—classical product development—it would be easier to convince someone to invest since the plan is clear and outlined. The lean startup approach does not have a clear plan which will make it more difficult to get the investors’ trust in an initial phase. Instead of investors entrusting their investment in a process of developing one single and defined product, which is then released into the market and is expected to generate revenues, in a Lean startup they will have to trust in a plan that is constantly changing and being tested, which for hardware does not seem very reassuring.

In terms of relying in validated learning to prove or disprove the company’s assumptions (i.e. validating them through empirical data from customers using the product) Prof. Jesus thinks it would be possible. However, he was of the opinion that building an initial MVP—which in his case he had the idea that it could only be a prototype—would imply higher costs, meaning higher initial costs in an initial testing phase. “**Regarding the sensors, we trusted in what our customers said they would need according to their medical problems, otherwise it would have been more costly for us**”. Prof. Jesus had the opinion that if they had followed a lean startup approach (by performing experiments to find out whether for example, a particular sensor would be required or useful or to measure the customers’ behaviour) then the startup’s costs would have increased substantially since several beta physical-products obviously do not bear the same costs as beta software-products.

As to the fast iteration cycles, Prof. Jesus thinks that they are possible in hardware development, however not for all cases. He thinks that if one is able to use cheap open source microcontrollers in their products—**he was using Arduino**—fast iteration cycles should be feasible (through firmware updates). However it would not be close to a perfect solution as there would exist problems unsolvable by firmware updates (as seen in 3.2.1.3). But if open hardware is not an option then things get much more complicated and difficult, according to Prof. Jesus.

Alteraid was constantly trying to reduce costs, and fast iterations with hardware would be inevitably more costly. Prof. Jesus is convinced that the reason why their cycles took so long was not a question of technology but rather a question of available staff dedicated to developing those improvements or new solutions. Therefore, the solution to accelerate their iteration cycles would be more economic resources invested in more staff.

Relatively to the reduction of batches size, when improving or creating new products the work unit was equivalent to: all the changes required to be implemented, tested and designed, in order to fulfil the goal they wish to achieve, which was approximately equivalent to 1 month of work. In the case of Alteraid, and probably in any hardware development case, according to Prof.
Jesus, there was not a culture of creating only minor improvements and test them immediately, especially because of Alteraid constantly aiming to reducing costs. Also, continuous minor improvements on a physical level imply higher costs and many other complications.

Finally, something that was uncertain to Prof. Jesus was the applicability of the Lean Startup methodology to any company (startup) regardless of its financial strategy and stage: bootstrapping, business angels or along the different stages of an external investment.

3.2.1.3. Techniques

If Prof. Jesus had applied the Lean Startup methodology in his startup, a prototype would be the minimum viable product that he would have built. This choice was especially due to how relatively simple and easy it was in his case to develop one. “The mockup for example would not easily convince customers. The client needs to use the product.” Moreover, as a physical product, the customer needs to actually use the MVP otherwise some problems that are inherently related with the fact that the product is physical and interacts in one way or another with its surroundings, might not be discovered. Prof. Jesus added that when installing the sensors at a client’s home, several problems came up. With a physical product, releasing the product to the customer's environment is therefore critical. An idea might be very appealing and seem easy to deploy, but the implementation in a real scenario may bring up different problems which probably will change the initial idea in some manner. “Initially we thought of selling an install-yourself product package with all the sensors and the gateway, but in the end we understood it was not possible, for instance we were unable to build an universal sensor that would fit in any door; so finally we had to make some changes in our product and in the company’s operations”. For physical products, in order to get valuable customer feedback before scaling the business, Prof. Jesus had the opinion that customers need to use and interact with the product. Otherwise costs would be considerably higher due to the need of dispatching technical teams to the customer's premises in order to solve problems that were not considered before and could have been avoided.

For Alteraid, the product (sensors) needed to be installed in each customer’s home and, according to Prof. Jesus, there were always particular problems related with the sensors’ installation and its spatial disposition relatively to the gateway and the patient’s preferred home-spaces. Therefore, for Prof. Jesus other types of MVP than the prototype itself did not seem to be viable.

For Prof. Jesus, in order to adapt the continuous deployment from the Lean Startup method to hardware development (through the already known-to-be-effective rapid prototyping techniques: 3D computer-aided modelling, injection moulds, and shifting the hardware product's value to software), the solution might be shifting the product’s value to software. However, it is not a very adequate solution due to the existence of some problems that are not originated in a problem of the product itself. Since the sensors’ microcontrollers are
programmed by the Alteraid, modifying and improving their firmware could be a solution for deploying continuous product optimizations and faster iteration cycles. However, they are not able to make corrections remotely because that would imply more complex and expensive microcontrollers in the sensors.

Regarding the error alerting and correction, as Alteraid is working with telecommunication devices, sometimes the problems are related with non-optimal spatial disposition of the sensors/gateway or with the existence of obstacles causing interference. Additionally to that, in order to have a faster and less costly testing phase with customers, Prof. Jesus used more standard sensors that were already available in the market and without some specific features. Moreover, according to Prof. Jesus, if the product consists in just one device, instead of several as it is his case, then continuous deployment could probably be better adapted and applied.

Regarding the use of split-testing technique for testing product optimizations in hardware products, in Prof. Jesus’ view it is something achievable but probably needing adaptation, and inherently more expensive and complicated than in software products. In Alteraid this technique was actually used but only after the business concept was validated. They had different group of customers using distinct sensor versions, and in this way they were able to understand which of them had better performance. For Prof. Jesus the iteration cycles build-measure-learn that these split-tests foster, are not adequate in the beginning but in a more advanced phase: “For me in hardware development split-testing has sense once the concept has been validated, not in the beginning.”

### 3.2.2. Fractus Interview

Fractus is the world leader in fractal antenna IP and licensing. Previously it was mainly a B2B company, but now is mostly a technology company, positioned at the very beginning of the value chain. Fractus designs, manufactures and licenses optimised antennas for mobile handsets, short-range wireless devices and telecommunications infrastructure by applying the science of fractal mathematics to antenna development.

Fractus holds an Intellectual Property Rights portfolio of more than 50 inventions protected through more than 200 patents and patent applications in the US, Europe and Asia. As a result of the impact of Fractus’ technology in the global telecommunications industry, Fractus has been named as one of the most innovative technology companies of 2006 with its inclusion in the top 100 private companies in Europe and the Middle East by the Red Herring Insider’s Guide and as a Technology Pioneer in 2005 Davos’ World Economic Forum.

The space-filling and multi-level properties of Fractus’ technology enable Fractus to achieve increased antenna performance and/or reduced antenna
size with optimum multi-band functionality. Fractus provides a high standard of customer service and technical support for antenna design engineers, project and programme managers and antenna purchasers worldwide to ensure development and production of optimal wireless devices, with more than 20 million units shipped worldwide. Presently it is developing technical solutions, however it started by developing and selling base stations and mobile handsets antennas directly to customers; it is that flavour, of an antenna developer and seller company, that was assessed in the interview.

Board of directors: Ruben Bonet, Founder and Chairman of Fractus
Dr Carles Puente, Founder and Chief Scientist of Fractus
Ron Epstein, CEO of Epicenter IP Group
Antonio Cuesta, Nexus Consultores
Cristian Fernández, Director at ICF Institut Català de Finances Capital
Marcel Rafart, Founder and Partner of Nauta Capital

Website: [http://www.fractus.com/](http://www.fractus.com/)

Address: Avda. Alcalde Barnils, 64-68 08174 Sant Cugat del Vallés, Barcelona (Spain)

3.2.2.1. About the company’s case

In the case of Fractus, the company was quite positive about how well the market would accept their product because of the performance improvements and other positive details of their antennas relative to those available at that moment. Therefore, Fractus looked for specific players in the telecommunications business field and presented them with prototypes from which they got valuable feedback. Hence, their assumptions regarding the customer hypothesis and the product were validated directly with the customer.

The steps taken by Fractus in the presence of a new idea, or performance improvement to be translated into a product update were the following: to engage with the potential customer, presenting them the product and getting feedback; and then to develop a prototype, test in laboratory, and run tests in real-life scenarios. However, according to Dr Carles, a new product it might require an engagement of several partners in order to convince the next player in the value chain, who in turn would need to check with their client. Therefore, new products typically required getting all the value chain engaged.

For Fractus, product optimizations were indeed common, however in a development- and test-phase only, since the cost and logistic complexity would have made it unfeasible to have continuous optimizations unless the improvements were worth the costs. “In hardware the product is not launched before as much optimizations as possible or desired have been achieved, because it involves manufacturing, maybe investments in making moulds and in having hardware in factories to specifically deal with that product manufacturing.”
The type of product that Fractus was offering, i.e. microchip antennas, did not leave space-left for problems to exist, also given the nature of the product itself. That is because previously to putting their product into production, they would do a thorough product testing and analysis. Therefore, issues and defects were rare to exist in the product sent to the customer. “In hardware it is normal to run a lot of tests before the product is finalised and released. You do it yourself and also with the customer. That is part of concurrent engineering which is used nowadays by several hardware companies. In case there was a problem with the product, the customer would inform it to Fractus and accept it. According to Dr Carles, Fractus was not able to detect problems in their products once they were being used by their customers, unless they received a complaint. Moreover, Prof. Carles did not see how those issues could have been monitored in real time or whether that would even be justifiable.

3.2.2.2. Applicability of the Lean Startup approach

Regarding the applicability of the Lean Startup approach to Fractus case and to any hardware products business in general, in Dr Carles’ opinion the main problem is the speed that this approach uses as one of its crucial characteristics. “It’s too fast.” If that speed is essential and possible in software development, for Dr Carles, such is not feasible in hardware development, whereat he is certain that the methodology needs adaptation. “The concept of this approach has very interesting aspects for any industry. How one can integrate these concepts into the company’s business model, can vary a lot depending on the industry.” Generally speaking in hardware development, the speed problem is, not only but also, related with the logistic requirements implicit in hardware development, namely the adaptation of the machinery in factories in order to deal with each specific product. Also another problem Dr Carles identified was the cost that such numerous fast iteration cycles, with product changes, represent.

In terms of the empirical data used to prove or disprove the company’s assumptions, acquired from customers using the product, for Dr Carles it was something definitely possible and ideal in hardware development. However, he did not see how that could have been done or even useful in Fractus particular case, because their product was fractal antennas to cell phone and mobile base stations. That only highlighted the need for adjustment that the method needs in order to be applied in hardware product development; “Hardware comprises many industries, each one as if it were a separate universe in most of the cases.”

For Dr Carles, fast iteration cycles in hardware development, and namely in his case, were definitely more difficult and challenging. However, according to him, they are clearly a desire in the industry of hardware development, since the periods of time implicated in designing, producing, testing and licensing (according to several regulations, for instance, UE safety directives) typically quite long.
Relatively to the possibility of having small work-batches in hardware development, Dr Carles thinks that it would widely depend on the industry. From microchip antennas, to boat production, the work that is passed between different departments (stages) must obviously be very different in terms of complexity and in the time that it requires to be concluded. In the case of Fractus the work-batches were equivalently large, never less than 3 weeks, on average. According to Dr Carles there were already similar concepts that were being applied in hardware development, namely: concurrent engineering, which allows parallelization of tasks; being flexible to change; and providing different alternatives so that customer can give his/her opinion and test the product.

3.2.2.3. Techniques

Regarding the minimum viable product for Dr Carles the mockup was very usual and typically was quite useful. It provided what is capable of driving the attention of the customer (i.e. the unique value proposition), and according to him, most of the times that is something tangible. It could be a combination of things as, for instance, a picture and the scale model, but having something tangible would help a lot, according to Dr Carles. Unlike software, hardware comprises several universes of hardware products. Hardware for mobile phones is an industry, hardware for cars is a different industry, and hardware for airplane companies is a completely different industry: "A car needs 3 to 4 years to be developed, a phone around 9 months, an airplane or a ship is developed in 10 years." Most of the times, according to Dr Carles, customers cannot anticipate their own reactions, whereby the problem with hardware is that the development effort is required to test and have those reactions. The complexity with software is equivalent to hours of code and it is easier to find shortcuts in order to make it work or make it look like it works. Therefore, for Dr Carles MVPs must be either prototypes or mockups. However, other MVPs as computer models for example, may be used to enable faster iteration cycles with customer insights, and feedback in a more initial phase of product development/designing.

Regarding continuous deployment, Dr Carles had the opinion that it does not make sense to think of continuous improvements and changes in hardware products. “You cannot continuously evolve the product as fast as you do software, because, for example, there are many parts that need to fit together and a change in one part will probably require more changes in other parts.” That is because the nature of the products—which implies higher costs, time, and complexity than software—does not allow to have continuous improvements and require a more intensive testing and commitment to achieve something close to perfect before sending it to production or presenting it to potential investors. Most of the times hardware product development involves several entities responsible for example for, quality assurance, safety directives, environmental directives, etc. “When developing hardware, the customer needs to test performance, therefore the product needs to have passed through extensive testing and improvements”. Nevertheless, for Dr Carles, in some cases and with the right adaptation, hardware should be susceptible to accept the changes that Lean Startup proposes in the ways of reaching the solution.
customers seek, enabling more flexibility in the final product, (e.g., different versions), so that it becomes possible to make changes depending on the customer reaction. However, the adaptations would vary between industries.

As to using the split-testing for product optimizations in the case of Fractus, Dr Carles did not believe it to be applicable, namely because it is not attractive for customers. “You have a small number of customers who are willing to test the product; therefore you cannot play with them. They will not be willing to give their time for free.” In Fractus, they commit to a customer, engage with them and do what is best for that customer. Moreover they may use different solutions for each customer, meaning that each customer gets a personalized product adapted to their needs. Therefore, the split-testing for proving or disproving product changes/improvements would probably not fit Fractus and probably similar companies.

3.2.3. On-Sun Systems Interview

Interview made to Dr Andy, from On-Sun Systems. On-Sun Systems is commercialising a disruptive solar concentrator photovoltaic (CPV) technology. The solar panels they sell have superior efficiencies and energy density. Based on the principal of focusing sunlight onto highly efficient multi-junction PV cells, On-Sun’s patented technology enables solar installations to generate substantially more energy per site. To the installer, the panels are flat, fixed, stationary devices, just like standard crystalline silicon panels. However, internally to the panel, the sun is tracked in order to maintain an optical concentration ratio of over 500 times.

As with all concentrator photovoltaics applications, On-Sun’s panels are ideally suited for installations within the Solar Belt. This includes (but is not limited to) the major markets of southern Europe, South West USA, India and China. On-Sun Systems has a finance office in the UK near Ledbury and a product development centre in Barcelona, Spain.

Having gained a PhD in Optoelectronics from Oxford, plus MBA from IESE, Barcelona, Dr Andy has worked in 3 startup companies in a variety of roles including product development, manufacturing management and project leadership.

Website: [http://www.onsunsystems.com/](http://www.onsunsystems.com/)

Board of directors: Giles Clark, Chairman
Andy Tomlinson, Chief Executive Officer more
John-Paul Szczepanik, Chief Technical Officer
Greg Finn, Chief Commercial Officer
Henry Hyde-Thomson, Director
Fred Edenius, Director
3.2.3.1. About the company’s case

In the case of On-Sun Systems, Dr Andy confirmed that they did try to test components, make low cost versions and validate aspects of the products and then try to put the whole design together. As to validating the company’s customer and product hypothesis, Dr Andy clarified how his case was different comparing to the former interviews where products actually are new to the market, in terms of technology or service offered. “The market for our product is very well defined: How big is it? How much does it cost? And how much electricity does it make? When we speak about more revenue streams, validating, incremental features, there are actually very few opportunities.” On-Sun systems actually used lean manufacturing techniques; however the company’s vision of what exactly the product needed to perform was quite clear. For On-Sun Systems there was only one trade-off: efficiency versus costs. In terms of interactions with the customer and market research, they did not do it very much. Dr Andy also pointed out that in their case, and as it is typically in all hardware companies, they were not initially trying to convince customers, but investors.

The company participated in several tradeshows, where it was able to show their prototype along with a sheet filled with product-performance results. Such sheet was obtained from a series of computer simulations, and was intended to represent the real performance of the product. The company focused onto achieving those values with the real panel, because they were exactly what customers cared about. In the exhibitions Dr Andy was able to speak directly to customers, some of those customers were whiling to buy sample panels. Once they agreed to buy the sample panels they had to agree to share the data they would collect from the panel (internet-connected solar panels) and a certain amount of minimum testing. “We currently have a good amount of customers lined up in this scenario, and after the testing is completed hopefully they will buy the product.” However, this whole process is very slow because customers have to: look how the product performs in different conditions, and to wait until there are enough weather conditions through different seasons.

In the presence of a new idea, or performance improvement, in On-Sun Systems, it usually began in the R&D. According to Dr Andy, it typically would begin as a design change, go through a series of tests and then it would be implemented onto a full prototype. After which they would commit the changes to production. The next phase would be re-qualification of the product (quality certificate), which was done by an independent organization. “We try to do something that functions in the exactly same way as the final product will to see if the customer will accept. Then there is an R&D project which takes the...
concept and actually makes it into a low cost solution.” The whole process would usually take up to between 8 and 12 months.

Regarding product optimizations, they were somehow frequent. However in the “back-office”, i.e. before launching into production, different product versions with product changes/improvements (several little changes) were made. Only after enough changes were worth making a big change, and also together made a significant difference in the product, then did On-Sun Systems send the new product into production. “We have generation of products, and each generation comprises several changes relative the former generation.”

As to product issues, On-Sun Systems implemented a monitoring system on the product, which enabled the product itself to self-report if there was a problem with it. “We would like the customer to let it report to us, it would give us valuable feedback. The company doing the maintenance of the product will be looking at the data and evaluate a possible site intervention.” Inside the panel there was a micro controller which controlled the motors and could also communicate to an internet-connected hub, giving the state of every panel in real time. No physical presence was required in the site in order to gather the data from the panel.

3.2.3.2. Applicability of the Lean Startup approach

For Dr Andy, the main and obvious problem was how fast iterations are in Lean Startup applied to software startups. “When we develop hardware it is typically on quite a long cycle time, the components for example take 2 months to arrive therefore there is a limit to how agile you can be.” For Dr Andy, fast iteration cycles would be possible in On-Sun Systems but only internally. Fast iterations, but in a phase preceding the new product’s generation release. It would never be something that would be showed to customers, according to Dr Andy. However, in their case of solar panels, even mockups or prototypes could take quite long to build.

In terms of obtaining empirical data from customers to prove or disprove the company’s assumptions, Dr Andy did not see how it would be applicable to his case. What On-Sun systems did was to participate in several exhibitions where they had direct contact with potential investors and customers, which allowed them to have feedback on the product they were presenting. Having the product and its performance results in display, they had a specific public with knowledge in terms of what was better and worst regarding the product’s quality and performance values, evaluating their product and giving out valuable feedback. However, according to Dr Andy, the market of solar panels is very well defined which made it simpler regarding the validation of both their product and customer assumptions.

Relatively to the reduction of batches size, when improving or creating new products the work unit, for example to have a design change, is equivalent to 4 or 5 months. “If we need for example to change the lenses design, it typically takes 6 weeks to redesign it, 8 weeks to get it made, 2 weeks to test it and then
if they want to do a product build on the basis of that it might take 6 weeks to do that build.” Therefore, for Dr Andy there was a limit to how small those batches could get, and they would probably never get even close to the size of the batches in Lean methodology applied to software.

As to the possibility of having real-time problem-monitoring and problem correction, given the nature of the product—by law, solar panels must endure and work for at least 25 years on the site where they were installed—this was clearly not possible in terms of hardware issues. “In a software environment, because you can do upgrades in the field, you can have software self-updates over the internet for example, whilst our product once is sold it is gone, is not going to be touch and must work for 25 years.” However, On-Sun systems implemented in their panels a technology which allowed them for a constant monitoring of some specific values, through an internet connection, namely the state of the motors. However, in terms of self-repairing, only software bugs would be fixable remotely, which Dr Andy led clear that they were not supposed to exist. Therefore, the desired problem-monitoring from the Lean Startup approach, is partially achieved, though for solving those problems a team must be dispatched to the site, which would typically take between 2 to 5 days.

3.2.3.3. Techniques

Regarding the use of minimum viable products, Dr Andy explained how in his case he actually used this concept, through the use of several cheap prototypes, mockups, and even smoke-tests as mentioned previously regarding the datasheets with performance results. “We made different sizes of panel which enabled us to prove the performance, and to show it to the customer at a low cost. We were able to make small prototypes very easily, cheaply and quickly.” With the prototypes, Dr Andy was able to show in the exhibitions the small solar panels’ prototypes to interested potential clients. For the customer, from the small prototypes it was easy to see how a full size solar panel would perform, because it was only a matter using the mathematical rule-of-three. The datasheets were used to lure customer attention and validate some of the assumptions that On-Sun Systems had made initially. Also, for the purpose of passing legal tests of product liability and exhibiting, full size mockups were used because it was the frame/chassis that had to endure the tests. “We always used datasheets that, even if simulated, were a quite accurate estimation of what the product’s performance would look like. We have also shown empty caseworks, a product which has got the external dimensions and shape but it doesn’t work.”

Regarding the continuous deployment, Dr Andy made clear that On-Sun Systems already used rapid-prototyping techniques in R&D. Those techniques have helped them quite a lot speeding up the process of product development. They not only had used injection moulds, but also had shifted part of the products value to software, by creating the “real-time monitoring system”. However, in terms of having continuous product changes/improvements and quick problem-rollback, for Dr Andy, that was clearly not possible. Not only because of the product’s nature, but also because a product change would
imply a quite long path that needed to be followed, each (physical) change, from design through product manufacturing, until legal certifications. “Continuous improvements for hardware can be possible, but not in production, and at a much lower than in software.”

As to the use of split-testing for testing product changes in hardware products, Dr Andy had the opinion that it would not be easy to apply it, at least due to complications with product quality guarantees. “The manufacturing people would tend to resist it because they could no longer guarantee product quality. A new certificate for product quality would need to be obtained, and also the tooling in manufacturing would need to change.” All the process is clearly very costly. Therefore, for Dr Andy seemed to be unfeasible to actually make product changes (design or performance) when there was not a clear idea if the customer would value them and would be willing to pay the extra price. However, for other scenarios if validation can be easily obtained and if the cost of making changes is practically absent, Dr Andy added that split-testing probably could actually make sense and be used for hardware products.

3.2.4. Technology Assistance BCNA 2010 Interview

Interview to Mr Javier Cañete, CEO of Technology Assistance BCNA 2010 S.L. (also known as TAB) startup. TAB is a small-medium enterprise founded in 2010 in Barcelona after a very successful European Project ‘Smart Sensors Systems Design’ with the aim of designing, developing and promoting new measuring and sensing technologies, based on novel and advanced methods, for measuring signals’ frequency-time parameters.

One of the main objectives of TAB is to reduce the existing gap between research and industry by introducing a new generation of cost effective smart sensing technology. Moreover, TAB also intends to establish their digital measuring technology and its advantages as an alternative to common analogue measuring principles.

TAB is dedicated not only to continuous enhancement of its existing product range, but also to responding to customer demands and suggestions with a high degree of flexibility and openness. This ensures that all solutions are custom tailored and represent the state-of-the-art in ultra-precise measurement technology for signals’ frequency-time parameters.

Website: http://www.techassist2010.com

Address: Parc UPC-PMT, RDIT-K2M building
c/ Esteve Terradas, 1, office N. 113 b
08860 Castelldefels, Barcelona (Spain)
3.2.4.1. About the company’s case

From Javier Cañete’s point of view, the Lean Startup approach was in fact favourable and adaptable to TAB. Their product, microprocessors, was characteristically very scalable and adaptable. The startup had adapted their product several times in order to meet their customers’ needs and requirements. “We have adapted our product several times to fit different markets: mobile phones, wireless technologies, sensors, etc.” Moreover, the startup was capable of a very short time to market, according to Javier, even though they had outsourced the chips’ manufacturing.

Therefore, the company takes very seriously and looks forward to answer to the specific requirements of each customer. According to Javier, in TAB the product was built and sold accordingly to what they understood it best suited the customer’s needs. Afterwards, the customer could request the same product (i.e. the same microprocessor but, for example, working in different temperature conditions. This request would quickly be taken care of and a new product version would be sold to the customer. Given how easy they were able to answer customer’s requests, Javier added an interesting comment regarding their concept of Minimum Viable Product: “In some sense we could say that our MVP is actually the fully working microprocessor since it is cheap, fast to develop and build.”

Technology Assistance BCNA is a business-to-business company, i.e. it sells microprocessors to other companies who will, in turn, embed the microprocessors in their devices (i.e. their products). If there would be an issue with a malfunction of the microprocessor the process would flow quite simply and predictively: “If there is a problem with our product, our customer contacts us and with their feedback we make the necessary changes on the product. Afterwards we ship it to our customer, in order to replace the broken chip in their hardware device.”

Regarding the process of validating the startup’s assumptions, the startup had proven they were tackling and existent problem in a small group of important customers (companies) who gave them very good feedback and even recommended their product to others. Javier made clear that they were offering a new line of product, whence they had only indirect competitors. TAB was in the process of understanding to which customers their product offers the most value. However, an issue for Javier’s startup in validating their assumptions was that the product required a lot of time before it was embedded in a device and actually used by the end customer. That translated into a difficulty for the startup to quickly validate its assumptions and get feedback from the end customer. “First our direct customer needs to make prototypes, once our product has proven satisfactory to them then they buy us more microprocessors, in order to implement them in new devices, and finally sell them to end customers.”

As to the metrics, the startup was still in the process of defining the metrics that were important to monitor. At the time of the interview, focus was being given to the number of customer that bought their products, i.e. the volume of purchase,
and also the number of units each company was interested in for the following year.

Regarding product optimizations, they occurred frequently. Those optimizations followed a sequence of stages: customer’s equipment analysis, feedback on the microprocessor’s pros and cons, in-lab testing, and deployment of the optimizations. “Our customer presents an analysis of their new equipment, they give feedback on which points our product could have better performance or be less precise; afterwards we validate which improvements are possible and we try to apply those optimizations in our lab. Then, after we have concluded all the possible optimizations we ship it to the customer.” The whole process would take up to 2 or 3 months.

3.2.4.2. Applicability of the Lean Startup approach

Javier has the opinion that the Lean Startup methodology might as well be a solution to drive down the startups' lack of success. “Only 5% of Startups in Spain resist beyond the first 3 years”. Also, Javier saw the Lean Startup approach adequate to TAB, and he believed that it would help them understand how they need to change their strategy in order to increase their products’ value-added and their growth.

In regards to fast iteration cycles, Javier was positive about their applicability to TAB given how easy, cheap and significantly fast it was to build the microprocessors. Nevertheless, Javier immediately found a problem: “The problem is the internationalization, due to the costly taxes applied in each country. We pay several distribution taxes; it is very complex to distribute our product.” Therefore, in order to apply fast iteration cycles, Javier had the opinion that they would need to have company headquarters in different countries and different number of iterations cycles in each one depending on the taxes. “Probably in Germany, supposing their taxes are more expensive than in Spain, we would do fewer cycles than here in Spain.”

Relatively to the reduction of batches size, in TAB they were already quite short, according to Javier. “The time required, which still is not close to a couple of days, is inherent to the hardware development process.” From developing until the microprocessor was ready to be released, it would take approximately 2 months.

Javier did not see the concept of real-time problem- monitoring and correction to be applicable to the microprocessors. According to him, it did not seem to add any value to the product, and because the product (microprocessor) was to be embedded in another device, it did not have sense of applicability. “It is impossible. Our customer has their own report and after they conclude that the problem is with our component, they send us their report.” However, regarding the ability to monitor/see how customers use the product (like it happens in software, through a cohort analysis) Javier had the idea that there were possibly other indirect ways which could work for getting the same or similar valuable feedback. “If we interviewed our customers regarding, for example, their
awareness about a side effect on a device they are using, which inside is using our product, then we could take conclusions on whether they were noticing that side effect. For instance, the increased battery lifetime of a mobile phone using our microprocessor.” From these indirect evaluations the startup would be able to know whether, in fact, that side-effect, which would be a result from a product improvement, was something that customers did value.

As to validating the startup’s assumptions with empirical results from real customers, surprisingly, TAB used something similar, according to Javier. They validated with customers who bought their microprocessors, who then gave back feedback. Moreover they also offered courses which gave an overview of their product and its benefits. “We also make courses to other companies in order to explain how they can use our product and how it adds value to theirs.” These conferences worked as perfect opportunities to validate assumptions that the startup had made regarding either the product or the customer. Additional to that, and also in order to try fighting the slow process of obtaining feedback (because they are a B2B startup and therefore it took extra time to receive feedback) TAB offered a way of both speeding up feedback and customer acquisition: “We have an evaluation board where our customer can easily and rapidly obtain results directly from it, validating whether it is according to their interests.”

3.2.4.3. Techniques

In regards to the use of minimum viable products, in Javier’s opinion it definitely was a concept needing adaptation if he was to apply it to TAB. That is because the nature and purpose of the microprocessors required them to be as perfected as possible before being sold. Not only their product is a critical part of another major device, but also that device typically would play a very sensitive role. Therefore, the microprocessor must be sold as perfect as it is possible. Moreover, once sold, the microprocessors are to be embedded in a board along with other electronic elements, wherefore Javier’s customer would not be interested in having prototype versions. “Once released, our product needs to be very precise and flawless because it has critical purposes where it is used.” However, again, also in Javier’s case and according to him, the use of smoke tests would definitely be possible in a more extensive way as MVPs. Actually, TAB did use datasheets. “I only see ourselves using MVPs if we think about smoke-tests, i.e. data sheets of the microprocessor and computer modelled images of the microprocessors, however I don’t see much added value in the latest.”

Regarding the Continuous Deployment, as it was understood along the interview, it is concept that apparently does not seem to have much applicability. The typical rapid-prototyping techniques, used in hardware to speed up the iterations, are not useful in microprocessors development process, according to Javier. Moreover, because of the very sensitive role of this type of product and the business itself, it did not seem possible to have product regular upgrades even if it had a software component associated to it. “I think it would be very difficult to have a continuous deployment strategy,
because we are a business to business company; after our product is ready, other company must develop new hardware with our product inside, taking a lot of time to insert our product inside their device and then finally release it."

As to the use of split-testing, Javier was quite positive about the concept because it could really provide them valuable feedback. According to Javier, the technique could clarify whether their products’ distinct features were valuable and desired from their customers’ point of view and it could be easily applied since the production/development process (time to market) was relatively fast in TAB. However, once more, because it is a business-to-business company it hinders the use of this technique. “Because of the time that it takes for our direct customer to actually use and decide whether he is satisfied with our product or not, it could be complicated to apply split-testing.” In order to take advantage of split-testing, Javier proposed that if their customers used the evaluation board then there would not be problem. With the evaluation board, according to Javier, TAB could take advantage of split-testing to validate its products’ different features. “If our customer used, for instance, our evaluation board then we would have faster feedback from different groups of customers with distinct product versions, and in that case split-testing would be valuable.”
CHAPTER 4. CONCLUSIONS

According to Ash Maurya author of *Running Lean* [3], the key to adapting the lean approach to any specific product is a clear understanding and separation of principles from tactics. The principles should be applicable to hardware; however the way of using them might turn out to be unfeasible or not adjustable in all extent. The inputs obtained from the interviewees were in accordance with that. The four entrepreneurs agreed on the possible and much advantageous applicability of the Lean Startup approach to their businesses. Naturally, being hardware much broader than software, the adaptation of the Lean Startup approach will evidently be different, from case to case. Hardware products comprise objects/apparatus/devices/machinery or a combination of all those, which one can actually touch, for instance, disks, keyboards, vehicles, antennas, etc. In contrast, software is untouchable, it exists as combination of symbols, algorithms and ideas—it has no substance. Therefore, it naturally is much simpler, cheaper and faster to manipulate than hardware products.

The interviews suggested that the principles behind the Lean Startup can and should be applied to hardware development. Generally, 3 steps probably can summarise how the Lean Startup approach can be applied to any product:

1. Firstly, one must document the business plan, a first version of it, using the specific lean tool for that purpose, i.e. the one page business model Lean Canvas;

2. Secondly, one must prioritize the riskiest parts of the plan and learn how to tackle those risks or adapt the plan, being that the starting point; and

3. Afterwards, one must systematically run experiments to test the plan, but experiments which maximize learning, speed, and focus on what are the real necessities and required actions of each stage, i.e. taking the right actions at the right time.

4.1. Findings

The valuable inputs given by the 4 entrepreneurs, and also all the information I was able to gather—from the supports listed in the bibliography—suggest a series of findings which I will hereafter enlist.

- The capital needed for the MVP will most probably be significantly higher in slow moving industries (hardware products) than in software development, therefore the “low burn start” from the Lean Startup approach might be difficult, or even not possible to achieve in some scenarios. However, in accordance with the Lean Startup, the investment is supposed to happen after pivoting, i.e. after having customers and a clear understanding of the desired direction to where the company heads
(what problem it solves and to which customers). As it is a physical product, it naturally implies higher capital needs invested, for instance, in several MVPs and split-testing. Having said so, it is understandable that it requires a higher early investment than in the development of software products. Nevertheless, a possible way of working around this issue might be to ask for real orders before having a real product. That is, asking for a purchase order and/or letter of intent in advance, and also creating scarcity by offering only a determine number of "early-customer slots", but charging them as true customers;

- As one of the key ideas of the Lean Startup, getting the customer involved with the product is essential, and for hardware that may be harder or less simple but it must not be avoided. Asking for feedback is not as effective as some might think. Customers truly reveal their preferences with their actions, whence the relevance of the MVPs;

- As it happens for software, for hardware the key still should continue to be iterating and connecting with customers early and often. But to apply that to hardware, prototyping seems to be the most natural solution. However, anything from a mockup image to a YouTube video of a prototype demonstration shall provide useful feedback. Moreover, product virtualization is also another tool which allows for deeper iteration with customers. Virtual prototypes, being interactive, enable integration with other components that may become part of the final solution, allowing to observe user’s reactions providing important opportunities to learn and to evolve, all before entering any manufacturing stage;

- As previously discussed, partially or entirely shifting the hardware’s value to a software component of the product may actually be the most effective way in order to speed up the iterations, and minimize cost and the time within the B-M-L loop. An example of that is, for instance, the use of flash-able and more complex microcontrollers, which by allowing firmware updates would be a viable solution for deploying quick improvements and corrections. The case of On-Sun Systems, who have implemented an internet-connected hub in their solar panels, is another example of this value shift; Nevertheless, it seems that such speed will be dependent on the type of product, since in some cases the amount of work required, between iterations, will inherently be smaller or simpler given the nature of the product or the role it plays;

- However, some products do not seem to have space left for a new added software component. That seems to be the case of microprocessors, for instance, which is a very well defined product in terms of the role it plays and the problem it solves. In these cases, given the fact that it is hardware it becomes very difficult to have a continuous accompaniment. Which, unlike in software, prevents us to take advantage of the possibility to monitor how the customer uses the product or what they find good or bad about it while they are actually using it;
• Testing different product features with different groups of customers, i.e. split-testing, seems to be tricky if applied in physical products development. While for some types of products may be feasible in terms of time and cost, and not risky as well, for other products it should be a very costly and slow process due to the high cost involved in manufacturing the product, the need of changing tools/machinery in factories, and more complex legal certification requirements. Because they are physical products, additional or removed features imply a different (brand-new) hardware product. Therefore it can be a costly process, unless the features’ value lies in an embedded software component of the product. Software is divisible, hardware usually is not, i.e., one cannot build just a part of the product that works. Therefore, it appears that for some markets, MVPs are cheaper and easier to be developed and built; on the other hand, for others it implies higher costs and complexity. Therefore, the timespan and cost of split-testing will probably vary between product types;

• In some cases, like in Fractus’, customers typically have different requisites which means that they will get unique and personalized products adapted to their case. The company will engage and commit with them. In that sense, it seems that either it is not applicable to use split-testing, or the risk in trying different approaches (different product versions) is high, especially when customers are in a small number. In software the magnitude of the changes done in a split-test, in terms of the risk involved, is quite lower;

• As physical products are not divisible, to come up with new ideas and deploy them right away does not seem to be a viable option in every scenario, unless in a pre-manufacturing phase. In such phase of product rapid prototyping, what is created is only a virtual product. The virtualization of the product in a first stage might be a solution so that the Lean Startup approach becomes more suitable to hardware products development. With hardware, simulation, virtual models, prototypes or emulation platforms can probably be used to demonstrate products, quantify progress and gather feedback;

• Therefore, fast build-measure-learn loops applied to physical products seem to have more sense (in terms of being actually quick loops) in a first phase of testing the product concept, with rapid prototyping and customer interaction, originating several virtual product versions. Here fast iteration cycles shall be possible and be as effective as they are in the lean startup method applied to software products;

• The direct relation between physical products and minimum viable products that suffer several changes in rapid build-measure-learn cycles is not direct and simple to foresee for a traditional entrepreneur. However, the path taken by 4 out of 3 of the interviewed startups (Alteraid, On-Sun Systems and TAB) to validate their initial assumptions, purports that those startups could most likely have followed an “adapted” Lean Startup approach;
• The feedback received from the entrepreneurs suggests that for hardware, unlike software, product optimizations would occur much more frequently in an initial phase and not when the product has been put available to all customers, as it happens with software/web-based-products. This is most possibly due to the cost and logistics involved when changing hardware products, which makes it unviable. Moreover, it seems that in hardware, optimizations make sense in bundle, i.e. instead of applying each optimization or improvement in production, apply them in groups of several improvements. Such improvements shall eventually make worth the cost and the time required to implement them in a new product version release;

• Also, according to the entrepreneurs’ feedback, there is a cost associated with hardware development related with all the prerequisites needed to actually develop those products (machinery and tools needed, licenses, etc.). Such suggests that this “anchored cost” will prevent the idea of continuously improve hardware products to exist, until the manufacturing processes get more Lean. Therefore, it seems again to make sense that all possible optimizations ought to occur in the initial phase before launching the product to potential customers. After that, every space for improvement will probably need to be stacked to other “product generation”, i.e. a new release with several changes and additional features;

• The feedback also suggests that real-time problem monitoring and fixing might only be partially possible. That is, if on one hand real-time monitoring might be possible through more complex and possibly expensive products (the product’s value is shifted, even if partially, to a software component of the product), and a continuous internet connection, on the other hand to fix problems immediately or revert the situation to a previous state seems to have no sense of feasibility. Those two actions, fixing and reverting, might as well just be possible in physical products which have embedded software. However, even in that case, there can be issues which are not related with the product itself but with its surroundings, and those cannot be solved remotely, without a team at the site or customer intervention. For some hardware industries, it seems there is no sense of applicability in having real-time monitoring of the products;

• Instead of directly monitoring the product and its effects on customers, in the cases where such is not feasible, it seems the solution might be checking for direct consequences on the product due to, for instance, the recent upgraded features. Or, for example, looking for a consequence on the way customers use the product. Thereby, entrepreneurs could also know, for instance, whether the new features their product has are desired and actually solve an existing problem, or are having the expected effect on the customer;
As the concept of hardware products (technological products) is wide ranging in type of products, i.e. type of industry, the concepts of Lean Startup will most probably be applicable at a different degree depending on the industry, mainly due to time-to-market constrains which can be widely different;

It appears that for physical products the main issues are cost and time-to-market constrains which directly work as barriers to the use of a fast iterations cycles. And that is more true for startups that are farther away from the end customer in the value chain (as it is the particular case of Fractus), because the need of engaging with the other players ahead of them increases. Therefore, it seems the Lean Startup approach fits best companies that are in the end of value chain. For them it appears to be more feasible to have fast iterations, to take advantage of direct contact with the end customer for feedback inputs and assumptions validation, and possibly to monitor the relation between the customer and the product. I definitely would not say that the principles are not applicable in the other cases, as the B2B companies, but for them the approach seems to require much more changes and might not be fully applicable, in some cases;

Moreover, in regards to how fast the iterations can get, it seems that for hardware development it not only depends on the type of product but also on the taxes applied which may differ from product to product in each country. That means that for startups it might not be financially favourable to have the same product (small) releases everywhere. This is something that in software business one may surely be restful about;

In hardware development, on one hand for some hardware companies several departments need to be engaged, on the other, for other companies several value chain players need to be engaged even if indirectly (e.g. researching the products that are being sold or products that are planned to be developed). However, transversal to almost all, regardless the industry, is the involvement of other organizations as, for instance, organizations that approve the product according to the regulations in place, e.g. quality and security standards, which slows down the time-to-market;

If on software development one can speak of small work-batches, as software is divisible and can be added up, on hardware development it seems that this concept becomes very difficult to implement. Again, it seems to have more sense, however, if we think of applying small work-batches in an initial phase of rapid prototyping, creating virtual products, interacting with the customer, in order to quickly test assumptions and avoid “building” or designing something that is not of the customer’s interest. Small developments, simpler customer validations, faster learning. After this initial phase, a manufacturing phase would follow, and there the concept of small work-batches seems to lose its feasibility, unless—and I am prone to think that is what needs to be done—there is a redefinition of what should be the typical right batch size for hardware
development. Given all the stages that a hardware product must go through while in a development/building process, and also the physical actions that it implies (e.g. moving the product from stage to stage, making tests, assembling, etc.), the batch size of work, in terms of time duration, will probably be different between distinct hardware products, but always considerably long, with a time unit of most likely weeks. Therefore, such reformulation of the work-batch size would need to be industry dependent;

- The shared experience of the interviewed entrepreneurs suggests that one cannot continuously evolve a physical product as fast as it is done with software, because, for instance, there are many parts that need to fit together and a change in one part will typically require more changes in other parts;

- Because hardware is not as receptive to monitoring the customers’ behaviours when using the product, the metrics that entrepreneurs are capable of assessing are significantly simpler and less versatile than those in software/web-based products. Contrary to software startups, where the most important metrics to focus on are the AARRR metrics (Acquisition, Activation, Retention, Referral and Revenue), in hardware it seems that they may vary greatly depending on each scenario, but the acquisition and revenue should surely be in the list. In order to validate their assumptions, entrepreneurs need to guarantee the product effectively and (preferably) efficiently tackles the problem that they promise to solve with their product, and afterwards get a number of paying customers. To be certain that such is actually happening, according to Eric Ries, a solution for physical products seems to be metrics which allow to correlate customer actions, i.e. actionable metrics, which help to analyse customer behaviour, demonstrating clear cause and effect. However, if the product is liable to have an embedded software component/part they then could probably be similar to the pirate metrics;

- It seems to be obvious that the earlier insertion of customer feedback (direct or indirect) also in hardware product development, is something not only desirable but favourable, in order to achieve the growth and (better) sustainability;

- The feedback of the entrepreneurs suggest that for some (physical) products, there is a limit of how agile the business can be in terms of customer interaction and fast iterations. The solar panels of On-Sun System are an example. Moreover, for some types of markets, as the market of solar panels, because the products are so well defined and have been used in a determined way for a long time, there is not much room to make improvements that justify the involvement of the customer or a further analysis of how the customer uses product. In order words, it seems that for products which do not allow customers to use their imagination/free-will when they are using it, then the Lean Startup principles, i.e. the Build-Measure-Learn loop, Validated Learning, and
Innovation Accounting, (not the principles behind Lean manufacturing or a Lean enterprise culture) do not fit as good or are not all required. This is supported by Dr Andy’s feedback on his first opinion on Lean Startup methodology if he were to use it in On-Sun Systems: “When you talk about customer validation... the market for our product is very well defined: How big is it? How much does it cost? And how much electricity does it make? When we speak about more revenue streams, validating, incremental features, there are actually very few opportunities.” For example the microprocessor, apparently it is a product that does not need as much initial customer intervention as another more-customizable product would in order to fit the customer’s interests. It seems that by being a very physical or, to say mathematical, product with a very precise function, a simple iteration with the customer shall be sufficient to develop the product and completely satisfy the customer. In other words, it is a product where there is not much room for differentiations, a product which answers concisely to a very concise problem.

The 4 interviewees were in agreement with the following: most of the principles behind the Lean Startup approach are advantageous to hardware development, comparing to today’s process of developing physical products, and needing adaptation to be used in each industry/type-of-product. However, something that was not clear was whether the agility and the continuous customer-intervention in the product development process, characteristics of the Lean Startup, make sense for every type of (physical) product. For several years the waterfall model seemed to prove suitable for many hardware industries. As seen, this model is based on the premise that design and construction of some product can be defined in advance through development of a detailed project plan. Afterwards, development becomes a mechanical process of staged execution, according to that plan. However, probably for the majority of the cases, this model has proven imperfect.

Whilst in software development, the act of designing and constructing software cannot be reliably defined in advance. It is a creative process and due to its inherent uncertainty it does not seem possible to accurately plan it in advance. Such unpredictability is created by a group of varying aspects present in software development, for instance, customer needs, target market, target technology and team dynamics. Therefore, it is agility and adaptability that drives success rates to a greater degree in software development, rather than the intense upfront planning.

Hence, the question that I believe it ought to be asked is:

- At what level is hardware development, for each situation (industry) a creative process or a rather defined one?

As mention previously, all the information I have gathered suggests that it considerably depends on the type of industry, i.e. the nature of the (physical) product. Probably some questions might be put in order to figure out the answer to the previous question. For instance, if the following are affirmative statements, then probably there is the necessary stability to classify the
development process as defined, and the lean startup, either does not apply, is not required, or applies but quite partially:

- Do product requirements remain the same over time?
- Do products consistently meet customer need?
- Are product architectures likely to remain the same over time?
- Would a product be designed and built the same way twice?

On a fundamental level, product development (both software and hardware) relies on creative problem solving. Successful products seem to be those that evolve over time, follow technology and life style trends evolution, and require real collaboration within a team which is constantly synchronized with the customers’ needs in other to effectively offer solutions to their problems. While there are obvious differences between software development and hardware development, there are also significant similarities. Hence, in general the benefits of the agility and customer intervention characteristic of the Lean Startup shall be advantageous to hardware development, and their techniques useful and suitable at different degrees, depending on how define or creative is the process of developing the product. An interesting result in my opinion, and next step would be to have an understanding of the Lean Startup “good practices” applied to each type or group of industries in hardware, i.e. physical products.

![Fig. 4.1 Sequential versus iterative development (adapted from [17])](image-url)
CHAPTER 5. BIBLIOGRAPHY


CHAPTER 6. APPENDIX

6.1. Adapted strategic planning model SEWOET matrix

After analysing the company and detecting the entrepreneurial characteristics, competitive advantages and weaknesses, the results are gathered in a matrix comprising the entrepreneurial-strengths, $S_E$, weaknesses, $W$, entrepreneurial-opportunities, $O_E$, and environmental-threats, $T$. The difference between this $S_EWO_ET$ matrix and the original and classic SWOT matrix is that entrepreneurial opportunities are used instead of the environmental opportunities, and both entrepreneurial characteristics and competitive advantages are both used in place of internal strengths.

For startups, it is critical the combination of entrepreneurial characteristics and competitive advantages with entrepreneurial opportunities [4]. Analysing the $S_EWO_ET$ matrix, we are able to extract the following:

- The $S_EO_E$ strategies used to push the company forward; the $S_ET$ strategies to face the environmental challenges;
- The $WO_E$ which help covering some of the internal weaknesses;
- And the WT intersection strategies which are defensive ones designed to reduce the harm from certain threats, although the best choice for startups in such scenario is to not enter the market.

<table>
<thead>
<tr>
<th>Startup SEWOET Matrix</th>
<th>Company analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment analysis</td>
<td>Entrepreneurial characteristics and competitive advantages</td>
</tr>
<tr>
<td>Entrepreneurial opportunities $S_EO_E$</td>
<td>$S_ET$</td>
</tr>
<tr>
<td>Threats $S_ET$</td>
<td>$S_ET$</td>
</tr>
</tbody>
</table>

Fig. 6.1 $S_EWO_ET$ matrix analysis [4]

6.2. Strategic Process Variables

The strategic process variables that collectively reflect how firms strategize, and that allow inferring about the effect of internal organizational processes on the relationship between a startup’s EO and its performance (sales growth) are the following:

1. Strategic Decision-making Participativeness: reflects how the firm’s major business-related choices are arrived at, as to the extent to which they are made through consensus-seeking processes versus autocratic processes, by the formally responsible executive;
2. **Strategy Formation Mode**: defined as the extent to which the firm’s strategy emerges over time versus being planned in advance;

3. **Strategic Learning from Failure**: defined as the firm’s self-capacity to identifying its strategic mistakes, the causes of those failures and the lessons rose from them.

Together these variables define how firms strategize, i.e. how they choose, learn from and refine their major business-related decisions and patterns they assume. There are many techniques that follow the participative decision-making approaches, for example: the nominal group technique, the Delphi technique, devil’s advocacy, and dialectical inquiry; they are all common group decision-making techniques which represent specific and structured approaches to participative management.

Although firms with EOs are more proactive, they will not be quickly responsive to new market opportunities if their decision processes are slow. Sometimes these participative approaches will be a slowdown to the firm, meaning that the highest risk and potential entrepreneurial opportunities may be rejected as a result of such approaches, limiting therefore the firm’s value-creation potential.

### 6.3. The Lean Principles

Originally developed in Toyota’s manufacturing operations, Lean is a continuous improvement approach that focuses activities on reducing wastes. Among Lean-Thinking’s principles are: the shrinking of batch sizes, inventory control, acceleration of cycle times, eliminating waste, amplifying learning, delivering as fast as possible, empowering the team and seeing the whole. All these are embraced by 5 major principles:

- **a)** Value;
- **b)** Value Stream;
- **c)** Flow;
- **d)** Pull Value;
- **e)** Perfection.

The starting point for lean thinking is the definition of value. It seems to be hard to correctly define value, this is partly because most producers want to make what they are already making and are good at making, and partly because customers, most of the times, do not know what they want ([1], pp. 15-90); they tend to ask for a variant of something they already have or are aware of existing. Like Henry Ford said: “If I had asked people what they wanted, they
Lean Management Applied to New Business Development

would have said faster horses;” the problem lies on starting in the wrong “place”, but also on the fact that value-creation often flows through many firms and each one tends to have different value definitions that suit their needs, which in the end do not add up. To correctly define value, a joint value analysis and a challenge of its older definitions are required in order to find out what is really needed (ibid.).

The first principle from the list above, a) Value, is expressed in terms of a good or a service, or even both at the same time, that meet the customer’s need at a specific price and specific time; it is ultimately defined solely by the customer—product engineers must not have influence in the matter but rather put themselves in the position of customer and try looking at the product or problem through the customer's eyes. Customers like products made to precisely meet their needs and to be delivered immediately, which introduces the concept of ‘where’ (the product is produced) inherent in the definition of value. Specifying value accurately is a critical first step in lean thinking, as providing the wrong product (good or service or both) even if in the best and most efficient way possible is still waste.

Secondly, b) Value Stream is defined as the set of required actions or activities to design, order and provide a specific product to the customer, from the very first step of obtaining the necessary raw materials to the last—deliver the product to the customer. There are three types of activities along the value stream; one of them adds value and the other two add “muda”—the Japanese word for waste (ibid.): (1) Value-added activities, which undoubtedly create value, for instance, welding together the tubes that form a bicycle’s frame; the (2) Type-one Muda activities, that create no value but seem to be unavoidable with the current technologies or production assets; and the (3) Type-two Muda activities, which also create no value and can be immediately avoidable, for instance, workers in a downstream activity stalled and waiting on an upstream activity. Firms completely managed in terms of mass-production thinking share this problem of a wasteful value stream, but the problem is not the competence of the managers who operate the firms according to an agreed logic; the problem is the logic itself.

One of the main reasons why so much waste has gone unnoticed in many—if not all—industries for so many decades is the lack of transparency (ibid). Firms involved in the product value stream, mainly for reasons of confidentiality, do not communicate with each other and therefore are not aware of each other’s work. Therefore, synergies toward the same goal (a good, cheap and fast product, ultimately resulting in a happy customer) are not created and “muda” emerges everywhere (ibid.). Thus, the lean thinking suggests the need for new firm-to-firm relations which foster transparency and a common goal: eliminate waste.

The principle of c) Flow is the basis for eliminating waste. It is defined as the progressive achievement of tasks along a value stream, making the product proceed without any stoppages or backflows, from stage to stage, since its very raw beginning until it is delivered to the customer. This opposes the traditional batch-and-queue way of thinking, which, it seems we are all born with. There is
an inherent belief that activities ought to be grouped by type (departments) so that they can be performed more efficiently and managed more easily; and performed by batches, i.e., first activities type A, then type B, and so forth. This “common sense approach” is undesired from the Lean point of view, in fact, whereas it keeps individuals busy and the equipment running hard, justifying dedicated high-speeds equipment and allowing individual workers to get automated and more efficient at their tasks, from the system perspective it is just the opposite: it is inefficient and waste-prone (ibid.).

To get manufactured goods to flow, the “lean thinking of flow” makes use of the critical concepts of the Just-in-Time production strategy—also pioneered by Toyota—which focus on reducing in-process inventory, and on continuous improvements on quality and efficiency; relying on a signalling system, Kanban (tickets or simple visual signals, such as the presence or absence of a part on a shelf), between different points in the stream, allows the upstream customer to know when to make the next part, hence avoiding product queues along the value stream. Ways to foster flow include: enabling quick changes over tools in manufacturing from one product to the next; “right-sizing” machines so that processing steps of different types (for instance, moulding, painting, etc.) can be processed immediately adjacent to each other, assuring continuous flow of the object undergoing transformation without stoppages and backflows; to ignore the traditional boundaries of departmental functions removing all impediments to the product’s continuous flow; and reducing work batches so that smaller jobs are processed which reduces queues and inventory.

As a result of converting from departments-and-batches to teams-and-flow, the time required going from concept/raw-material to launch/customer is dramatically reduced. Nevertheless, the flow principles can be applied to any activity and always the amount of human-effort, time, space and tools needed, will be typically reduced to half (ibid.).

Work should be designed around small units that have value to a customer, allowing to identify and address bottlenecks very quickly, and link metrics directly to flow, customer value, and profitability. Other factors that can slow the flow of value include:

- Management interference – occurs in systems in which clear processes are not defined and workers are not charged with the responsibility for their work;

- Separate departments – tend to function along batch and queue lines. Also, because each department has its own priorities, resources are applied inconsistently, resulting in waste;

- Approval cycles – generate obstacles to flow by requiring reviews of batches of products and requiring status and performance meetings; and

- Changing requirements –most examples of “unanticipated change” could have been anticipated and negotiated in advance.
Also of interest and some relevance is to mention the simple but very enlightening conclusions that the Polish psychologist Mihaly Csikszentmihalyi has reached, after investing twenty years on reversing the usual focus of psychology. Csikszentmihalyi concluded that the type of activities (tasks) which a person finds as most rewarding is that which involves a clear objective and immediate feedback on the progress toward it, a sense of challenge, no interruptions or distractions and a need for intense concentration—the task itself becoming the end rather than a means to something more satisfying as, for instance, money. According to Csikszentmihalyi, someone experiencing these conditions is in a satisfying psychological “state of flow”; writing a book, being part of an interactive game, for example, or to work in an organization where every employee has immediate knowledge of whether the job has been done right and can see the status of the entire system, are all “flow experiences”. However, classic batch-and-queue work conditions are hardly conductive to a psychological “state of flow” since workers are only required to concentrate and apply their skills on a portion of the whole task, often having no immediate feedback, and are constantly being interrupted in order to deal with other portions of other tasks inside their area of responsibility.

The fourth lean principle, d) Pull Value, allows customers to define what they want and then designing and producing something that meets what they want. Lean businesses do not push a product onto their customers; a product is created only when the customer orders it. Nothing is produced by the upstream supplier until the downstream customer signals a need (pulls the product)—each activity pulling the next; the production processes are tuned to levels of customer demand. This contrasts with the common pushing systems that are unresponsive to customers and result in unnecessary inventory growth (ibid.)—and inevitably in waste. Companies switching to a Pull approach see their warehouses immediately shrinking, as the amount of excess (“just-in-case”) inventory—called work-in-progress, WIP, inventory—is dramatically reduced ([2], pp. 184-206).

Finally, the fifth lean principle, e) Perfection, consists on the endless pursuit of the improving and maintaining the value flow, as there will always exist activities in the value stream that are considered muda, which with technology progresses may become avoidable—a complete waste-free value stream is rather an ideal scenario (ibid.). Measures to encourage the pursuit of perfection include: on-going staff training; reward systems for initiatives that improve throughput, customer satisfaction, and quality; innovation sharing; and cross-team competition. Systems that encourage the pursuit of perfection in Lean business tend to be cross-functional and horizontal in composition. For organizations in which a horizontal organizational structure is impossible, value flow managers can be employed to oversee value stream performance.

With those 5 key principles in practice results a Lean system. A Lean system which must exhibit the following characteristics:

- Customer focus – provides the framework in which disciplined work practices emerge, giving the customer the service or product they want at the right time and right price (just-in-time production). Customers’
needs and expectations provide the pull that puts in motion all the enterprise activities;

- Waste elimination – waste is any resource or process that a customer is unwilling to pay for. The seven identified wastes are overproduction, waiting time, inventory, movement/motion, processing, rework, and transportation. By eliminating waste, companies can cut costs and improve quality and safety;

- Value creation – value is what the customer is prepared to pay for and that what is valuable to all stakeholders. Lean thinking requires that businesses identify value and eliminate all that does not add value to a process or product;

- Dynamic strategizing – flexibility and adaptability to customer demands requires a dynamic, strategic approach to production and the provision of services. The continuous improvement based on the knowledge of all involved in the business is a fundamental problem-solving process used in Lean business. This approach is not about making one-off improvements or changes, but rather about constant reassessment, which allows on-going and incremental change that, in turn, increases value within the business; and

- Knowledge-based strategizing – Lean requires the ideas and effort of the entire workforce in identifying value and eliminating waste. Knowledge-based strategies recognize the critical role of people, so investment in training is crucial. These strategies, combined with the concept of continuous improvement, can dramatically improve production and quality.

### 6.4. Traditional vs. Lean Startup Comparison Summary

**Assume growth is by execution**
- ✓ Find what customers want before building it (not surveys);

**Assume problem, product features and customer are known**
- ✓ Assume customer and features are unknowns;

**Slow iteration cycles**
- ✓ Rapid hypothesis testing (Minimum Viable Product - MVP);

**Invest resources in planning and product perfecting before launch**
- ✓ Build low-burn companies by design;

**Traditional product development method (waterfall) and large work-batches**
- ✓ Continuous deployment:
Lean Management Applied to New Business Development

- Work in small work-batches; fast iteration cycles; revert a bad change quickly; immune system: automated tests, real time alerting, on a failure fix the problem and improve defences at each level;

**Test in macro scale**
✓ Test in micro-scale first, then scale-up;

**Functional departments**
✓ Two cross functional teams: Problem team and Solution Team (100% responsibility);

**Traditional accounting**
✓ Innovation accounting.

6.5. **The AARRR Metrics**

Also known as “Pirate Metrics” (because of the resemblance of the abbreviation to the sound of a pirate), they were built for the purpose of being used by software companies, however the model is applicable to many different types of businesses, namely to physical products businesses ([3 pp. 23-45]).

![AARRR Metrics Diagram](image)

**Fig. 6.2** Pirate metrics [3]

**Acquisition**: It describes the point when an unaware visitor becomes interested in the product/service. For example, in the case of a flower shop, getting someone walking by the window to stop and come in the shop is an acquisition event.
Activation: It describes the point when the interested customer has his first gratifying user experience.

Retention: It measures repeated usage and/or engagement with the product; very useful to measuring product/market fit (ibid.).

Revenue: It measures the events that generate customer payments.

Referral: Also part of the use acquisition channel, it means the happy customers refer potential customers and drive them to use the product.

6.6. Prioritising Risk

A way of prioritising risk proven successful is the following [3]:

Product risk: Getting the product right

1. First make sure you have a problem worth solving;
2. Then define the smallest possible solution (MVP);
3. Build and validate your MVP at small scale (demonstrate Unique Value Proposition);
4. Then verify it at large scale.

Customer risk: Building a path to customers

1. First identify who has the problem and would pay for a solution;
2. Then narrow this down to early adopters who really want the product;
3. It’s OK to start with outbound channels;
4. But gradually build/develop scalable inbound channels—the earlier the better.

Market risk: Building a viable business

1. Identify competition through existing alternatives and pick a price for the solution;
2. Test pricing first by measuring what customers say;
3. Then test pricing by what customers do;
4. Optimize the cost structure to make the business model work.

Fig. 6.3 Systematically tackling risk in stages [3]