Anh Nguyen-Duc · Jürgen Münch
Rafael Prikladnicki · Xiaofeng Wang
Pekka Abrahamsson *Editors*

Fundamentals of Software Startups

Essential Engineering and Business Aspects

Springer

Fundamentals of Software Startups

Anh Nguyen-Duc • Jürgen Münch •
Rafael Prikladnicki • Xiaofeng Wang •
Pekka Abrahamsson
Editors

Fundamentals of Software Startups

Essential Engineering and Business Aspects

Springer

Editors

Anh Nguyen-Duc
Department of Business and IT
University of Southeast Norway
Bø i Telemark, Norway

Jürgen Münch
Center for Entrepreneurship
Reutlingen University
Reutlingen, Germany

Rafael Prikladnicki
School of Technology,
PUCRS,
Porto Alegre, Brazil

Xiaofeng Wang
Facoltà di Scienze e Tecnologie
Informatiche
La Libera Università di Bolzano
Bolzano, Italy

Pekka Abrahamsson
Faculty of Information Technology
University of Jyväskylä
Jyväskylä, Finland

ISBN 978-3-030-35982-9 ISBN 978-3-030-35983-6 (eBook)
https://doi.org/10.1007/978-3-030-35983-6

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Foreword

Peter Drucker famously said that the purpose of a business is to create customers. Everything else is in service of that purpose, including R&D, sales, marketing, and general management. Customers, however, are constantly interested in either lowering their cost or receiving more value from their relationship with partners and suppliers.

Most established companies are fighting a continuous war to avoid or at least minimize commoditization of their offering as, in a commodity market, the only factor that matters is cost. The most effective way to achieve this goal is through innovation. Innovation can be classified as sustaining and disruptive. In our research and experience, we have learned that most established companies are quite good at sustaining innovations that maintain and evolve their existing offerings, but surprisingly unsuccessful when it comes to disruptive innovations.

This is where startup companies enter the picture. Defined as organizations that operate under very high levels of uncertainty, startups are the key mechanism that society as well as established companies has for exploring radical innovations.

In my experience, a wonderful analogy that defines startups comes from Reed Hoffman, founder of LinkedIn, who compares a startup to jumping off a cliff with all the parts that you think you need to build a flying plane and hoping that you manage to assemble all the parts before you hit the ground. The analogy extends quite far as it explains how even limited revenue early in the process can help extend the runway of the company as well as how subsequent capital injections can be viewed as balloons that bring the "airplane under assembly" to a higher altitude, so that it has more time until it hits the ground.

Although the term "startup" is hyped these days and many are calling themselves "entrepreneurs," it is important to realize that starting a business in an established domain and using a business model that is widely spread in that domain is different from building a startup company. The latter is, at its core, concerned with aiming to successfully establish a radical innovation and to create a business around this.

The typical startup has full autonomy in how it pursues its goals. This is critically important as any wisdom and experience shared by senior managers from established businesses is concerned with the way business used to be conducted and consequently does not represent the new innovations that the startup is looking to capitalize on. In the case of “intrapreneurship,” where innovators seek to start new businesses within the walls of established companies, this autonomy is often challenged as decision-makers easily fall back into traditional decision-making processes and an outdated belief system.

As startup companies and their founders are, because of the aforementioned autonomy, automatically alone and self-reliant, it is no surprise that many avoidable errors are repeated in startup after startup. This is because, despite the vast amount of content available on the web, there is very little available in terms of validated scientific material that founders and others can reliably use in good faith. The book that you are now holding is an answer to that need and in the following parts and chapters, you will find a wealth of knowledge concerning success factors and advice for startups.

Although primarily intended for founders and employees of early-stage startups, the book also provides invaluable content for educators, policy makers, those involved in incubators and accelerators, mentors, innovators, and those generally interested in the startup space.

The editors have done an amazing job pulling such exciting, high-quality material together in a book that is bound to become one of the cornerstones of the startup community. Congratulations to the editors and authors and I wish you, the reader, a great experience!

Start-Up Board Member, Angel Investor and Advisor Professor Jan Bosch
Chalmers University of Technology
Gothenburg, Sweden
August 2019

Preface

The Era of Software Startups Startups have been a steadily growing global movement in the last decade. According to an industry survey, global venture capital investment in startups was at a decade high in 2017, with over $140 billion invested in startups [5]. Total value creation of the global startup economy from 2015 to 2017 is at $2.3 trillion, a 25.6% increase from the period between 2014 and 2016 [5]. AngleList, a database of startups, documented 4,791,000 registered companies in 2019, many of which develop software as a core asset of their businesses. Software ventures such as Facebook, Snapchat, Spotify, Pinterest, Instagram, and Dropbox, to name a few, are examples of software startups that evolved into successful businesses with fast user acquisition and rapid growth.

Every great company started as a new venture, and every great product started as someone's idea. We all have good ideas for the next big things, but relatively few of us have the spirit, energy, and skills to develop those ideas and visualize them into commercial products. Among them, very few entrepreneurs become successful. The high-risk nature of startups can be traced back to team, market, finance, and product factors [6]. Many startups face the challenge of creating technologically innovative products with cutting-edge tools and techniques. Many others have difficulties in acquiring financial resources, obtaining paying customers, and defining appropriate business strategies to deliver actual value [6]. Such diverse challenges underscore the need to research on software startups, which will benefit both startup community and interested professionals.

Defining Software Startups The usage of the term startup (start-up) in the sense of building ventures can be found as early as 1976. In a *Forbes* article, it said "The unfashionable business of investing in startups in the electronic data processing field." The term became very popular during the 1990s and 2000s with the arrival of the dot-com business. In 1994, Carmel was among the first who investigated processes inside small companies building and marketing innovative software products [2]. In 2000, Sutton, in his article in *IEEE Software* journal, introduced the term "software startups," and the role of processes in these companies [16]. More recently, with the movement of Lean Startup and Customer Development, the

Fig. 1 Software startups and relevant terms

term “startup” is getting increasingly popular. Steve Blank described a startup as a temporary organization aiming to create high-tech innovative products without having a prior working history [1]. Blank differentiated a startup from a small and medium-sized enterprise (SME) business, which does not necessarily intend to grow and consequently lacks a scalable business model [1]. Eric Ries defined a startup as a human institution designed to create a unique product or service under extreme uncertainty [12]. Rather than a formal company, a startup should be considered as a temporary organization that seeks a validated and scalable business model [12].

Software startups and similar terms, as seen in Fig. 1, are widely used in various disciplines. To define the scope of software startups, we need to identify the role of “software” in their value propositions. For instance, Steininger defines four ways software relating to startups [15]:

– Software-facilitated: software is used to facilitate business activity without direct contribution to core business value.
– Software-mediated: software is used to connect startups to their clients, i.e., e-commerce.
– Software-bearing: software is used as a part of the company’s infrastructures and products.
– Software ubiquity: a company’s value creation completely relies on the software.

Software Startup Research Despite the popularity of software startups, many questions remain about the validity of software startups as a research field and startup education as a necessary academic subject. There is a widespread belief that startup scene is simply too heterogeneous with diversity across subjects, industries, cultures, nations, etc. and that much of the success of entrepreneurs could be attributed to innate personality traits. Stories of young, successful tech entrepreneurs who had dropped out of school seemed to support this opinion.

But over the past 10 years, the demand for and the interest in entrepreneurship research and education have grown significantly. Researchers and educators in the entrepreneurship fields appear convinced that entrepreneurship can be taught [8, 9]. Specific knowledge areas and entrepreneurial skills are obtained by learning, either from entrepreneurs' own experience or from others. An entrepreneur can come up with a great idea, but he needs to learn to develop and validate a sustainable business model. The entrepreneur can have various ways to approach funding source, but he needs to learn to convincingly pitch to his investors. And when developing a specialized service or product, the entrepreneur needs to know well what he will offer and in which way the product can be made in the startup's contexts.

The interest in teaching entrepreneurship has increased at all levels of education, from university to training programs in incubators and accelerators. Recently, new methodologies and tools have been created for entrepreneurs, for instance, Steve Blank's customer development methodology [1], Alexander Oosterwalder's Business Model Canvas [10], and Eric Ries's Lean Startup methodology [12]. There are also successful programs for entrepreneurs that include structured educational components, for example, Startup Weekend, NEXT program, and incubator programs. These tools and programs emphasize the empirical view on startup education—a methodology of getting out of the classroom and engaging with (and learning from) the market to build a business model over time.

In the last decade, there has been an increasing effort on research and education that covers the intersection between engineering and entrepreneurship. This effort can be identified via a Google search with key words "Software startup research." In particular, software startup research can be defined as the use of scientific, engineering, managerial, and systematic approaches to successfully develop software products in startup companies. Similar to software engineering, the areas of software startup research cover software engineering knowledge areas that are closely linked to the startup context, such as requirements engineering, prototyping, and management [17].

In terms of scientific or educational books, however, no book could be found that address software startup topics. Existing materials, such as books, academic journals, etc., typically focus on either engineering and technology aspects or entrepreneurship and business aspects. Particularly, entrepreneurship books often focus on entrepreneurial skills, personal trait, behaviors, and entrepreneurial process. A book about software startup research will address the scientific area between software business and software engineering aspects. It can cover how technical aspects are related to business aspects, which complications arise, and how they can be dealt with.

The Target Audience The book is designed for a wide range of audience, so that as a software engineer, an entrepreneur, or an educator, you will find useful content from it. Firstly, as a software engineer, you wonder why you should know something about software startup research?

1. Startup movement is an inevitable trend. Software startups are an important segment that stimulates economic development and produces new jobs.

Entrepreneurship is a driving force for economic growth and software technology is a business-enabler, both facilitating large-scale innovation. Software startup research is the area combining both of these cutting-edge frontiers of the modern world. As software startup engineers, you are in a position to be excellent wealth generators—not only for your new ventures but also for the companies you work for and the society you live in.

2. You are likely to be a founder of or to work for a software startup company. In the second case, your employer does not necessarily want you to quit and form your spin-off company in direct competition with them. But software companies are dependent on innovation and growth for survival, and it is the entrepreneurial spirit that drives this. Besides, you might work in an entrepreneurial environment that is much different from your previous working experience in a corporate. Knowledge about software startup research prepares yourself for working, achieving, and thriving in such an environment.
3. Do you want to work on someone else's project for all your life? As a working professional, you need an interesting, stimulating, and productive working life; this means at various points in your life you will want to set different kinds of ventures in action. You may or may not create your own business, but you would certainly have interesting ideas for software, projects, and practices that you would like to promote in your work environments.

Secondly, as an entrepreneur, is it necessary to know about software startup research?

1. You will need to know everything about your company. Early-stage startups are often limited to various types of resources. In the beginning, you would be the one who makes a business model, hire first employees, create first prototypes, and design marketing strategies. Knowledge about software and fundamental engineering principles can accelerate your jobs in the early phases.
2. A software startup is a mutual interaction and evolution of business and product development. From a managerial perspective, you should know very well how to develop your venture, but also how to generate customer value via the development of your core services or products. Given the central position of a software-intensive product, knowledge about software development, Minimum Viable Product, technical debt, etc., are important engineering-level knowledge.
3. An entrepreneur needs to be aware of various micro- and macro-environmental factors during his/her entrepreneurial journey. Many product and engineering factors play essential roles in the success or failure of a startup at a certain point of its life span.

Thirdly, as a software engineering educator, how do you find software startup research useful for your teaching?

1. There is an increasing demand for startup education, especially in the engineering sector. As a university lecturer, you might already see educational programs at the master's or even the bachelor's level that combine both entrepreneurship

and software engineering. You will need a reference source of state-of-the-art knowledge about software startup research.

2. Being a trainer, an educator, or an incubator manager, you are a key player in a startup ecosystem. It is necessary to know the overall picture and the connection among the ecosystem's elements and to manage toward desirable outcomes. One way to learn this is via similar experience reported by others.

The content of this book is aimed at those who are interested in understanding the connection between business and engineering aspects in the context of software startups. As shown above, people in various roles can find relevant content from this book.

The Authors The major contributions for the book come from the Software Startup Research Network (SSRN),[1] which enables interactions and collaborations among researchers and stakeholders in startup ecosystems. SSRN envisions to (1) spread novel research findings in the context of software startups and (2) inform entrepreneurs with the necessary knowledge, tools, and methods that minimize threats and maximize opportunities for success. As part of the network initiatives, the First International Workshop of Software Startups (1st IWSS)[2] was organized in 2015 at Bolzano, Italy. Since then several workshops, conference tracks, and entrepreneurial events were organized under SSRN's member initiatives. The network also accounts for 100+ publications in the software engineering area.

There are 41 authors contributing to 20 chapters of the book. Among them, 78% of the authors are from academia, and 22% of them are practitioners. Fifty-eight percent of the authors are members of SSRN. The authors come from different known research organizations including Norwegian University of Science and Technology,[3] University of South Eastern Norway,[4] University of Jyväskylä,[5] University of Oulu,[6] Free University of Bozen-Bolzano,[7] University of São Paulo,[8] Pontifical Catholic University of Rio Grande do Sul,[9] and many more (Fig. 2).

The Topics This book presents important topics for the engineering and management of software startups, as shown in Fig. 1. Startups typically end up with a product that is different from their original ideas. Pivots, strategic changes in business, and product directions happen in every startup. Chapter "Pivoting in Software Startups" discovered different types of pivots and factors that trigger them.

[1] http://softwarestartups.org.
[2] https://ssu2015.inf.unibz.it.
[3] https://ntnu.no.
[4] https://usn.no.
[5] https://www.jyu.fi.
[6] https://www.oulu.fi.
[7] https://unibz.it.
[8] https://www5.usp.br/.
[9] http://www.pucrs.br.

Fig. 2 Main authors' affiliations

It is crucial for startups to acquire needed knowledge, skills, and capabilities for product development. Chapter "Yes, We Can! Building a Capable Initial Team for a Software Startup" looked at startups from the lens of competence, i.e., team capacity and team roles. Another inevitable phenomenon in startups is technical debt. Under time pressure and limited operational environment, startups rely on short-term technical solutions. Chapter "The Perception and Management of Technical Debt in Software Startups" investigated how such software professionals perceive and manage technical debt in their projects (Fig. 3).

The core concept in Lean Startup Methodology is Minimum Viable Products. The concept is the most overused and misunderstood terms among startup communities and researchers. Chapter "An Analytical Framework for Planning Minimum Viable Products" offers a rich description of Minimum Viable Products in software startups and proposes a way they can be created. In a larger scope, Chapter "Software Startup ESSENCE: How Should Software Startups Work?" proposes a method and tools to capture startup ways of working.

Metrics are structured data in useful and systematic ways, valuable for various decision-making in startups. Chapter "Startup Metrics That Tech Entrepreneurs Need to Know" presents a list of 100+ metrics for software startups that are compiled by studying practitioner-oriented literature. The chapter illustrated the usage of important metrics in a case study. Chapter "Early-Stage Software Startups:

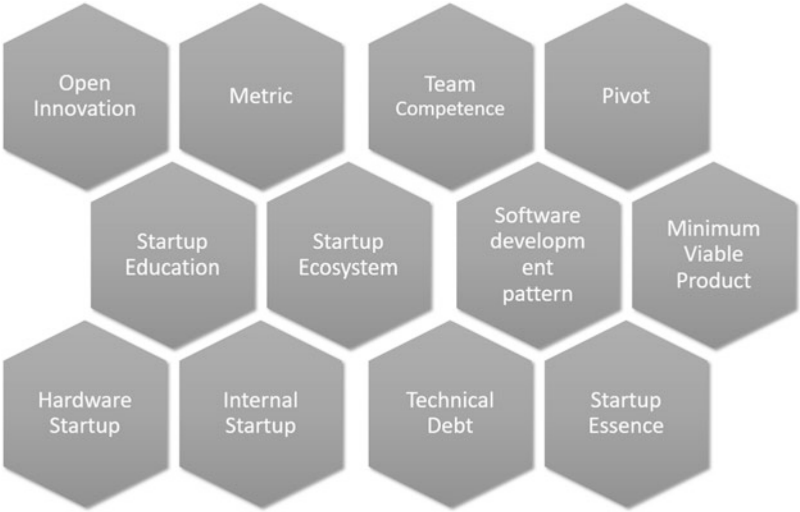

Fig. 3 Thematic map of the book's topics

Main Challenges and Possible Answers" summarizes common pitfalls and patterns observed in many startups.

Startups' operational environments include both micro and macro factors. At the macro levels, descriptions and evaluations of startup ecosystems in various countries are presented in Chapter "The Roles of Incubators, Accelerators, Co-working Spaces, Mentors, and Events in the Startup Development Process," in Chapter "Fostering Open Innovation in Coworking Spaces: A Study of Norwegian Startups," in Chapter "Startup Ecosystem Maturity and Visualization: The Cases of New York, Tel Aviv, and San Paolo," and in Chapter "Thailand's Software Startup Ecosystem."

This book also presents the educational aspect of software startups. Chapter "Software Startup Education: A Transition from Theory to Practice" reviews current literature on teaching startup principles for IT students. Chapter "Teaching Through Entrepreneurship: An Experience Report" presents an experience of teaching Lean Startup using a supportive platform. In the end, Chapter "Lean Internal Startups: Challenges and Lessons Learned" shares three stories of startups inside cooperates, i.e., how they adopted the Lean Startup model, their consequences, and lessons learned. Chapter "Software Startup Education: Gamifying Growth Hacking" reports a growth-hacking course for startups.

Besides empirical validation of software startup research methods and practices, the book gathers startup stories from Norway, Latvia, Russia, and Brazil. Chapter "Key Influencing Factors in Early-Stage Norwegian Hardware Startups: A Trilateral Model of Speed, Resource and Quality" describes the concerns with quality and agility in Norwegian hardware startups. Chapter "The Rise and Fall of a Database-as-a-Service Latvian Unicorn" reveals a postmortem experience of a falling software unicorn. Chapter “Triggers of Business Success of IT Startup Owners in Russia” shares local experiences of startups in Russian markets. Chapter “Brazilian Startups and the Current Software Engineering Challenges: The Case of Tecnopuc” describes the life of startups in a Brazilian technology park.

Methodological Foundation Many of the book chapters have backgrounds in empirical software engineering (ESE), establishing the methodological foundation for the book. Rooted in medical science, an empirical study performs a test to compare what we believe with what we observe [7, 11].

At the heart of ESE are empirical methods, which are adopted from more established research fields, i.e., medical science and social science. As a subset of ESE, software startup research involves significantly non-technical factors, both at the micro and macro levels. Hence, many of the research methods that are appropriate to software engineering are drawn from disciplines that study human, business, and society [14]. The fundamental principle is that our claims must be backed by evidence and such evidence should be based on concrete observations, measurements, or experimentation resulting from qualitative and quantitative research, often based on a multi-method or mixed-method methodology. These methods all have known flaws, and each can only provide limited, qualified evidence about the phenomena being studied. However, each method is flawed differently and viable research strategies use multiple methods, chosen in such a way that the weaknesses of each method are addressed by the use of complementary methods [3]. In each chapter, you will find how authors come up with their findings, and how they eliminate threats to the scientific validities of their work.

Many of our chapters ask exploratory questions, as they attempt to understand the phenomena and identify useful distinctions that clarify our understanding [4]. For instance, in Chapter “Yes, We Can! Building a Capable Initial Team”, the author asks: “What are the different types of pivots software startups have made during their entrepreneurial journey?”. Or in Chapter “The Maturity of Startup Ecosystems: The Cases of New York, Tel Aviv, and San Paolo,” the authors seek to answer: “How do coworking spaces foster open innovation practices?”.

As theories are underdeveloped in software engineering [13], many chapters in this book have the same style, as the frame of reference is expressed by the viewpoint of the researchers or practitioners. For those who are interested in a theoretical aspect, Chapter “Pivoting in Software Startups” offers an overview of popular entrepreneurial theories and how to use them to explain for software startup phenomenon.

One of the major references using in this book are the Silicon Valley practitioners’ methodologies, Lean Startup [12] and Customer Development [1]. We refer to terms, such as Minimum Viable Product, Pivot, and Customer Development, and explore them in specific engineering and business contexts.

Fig. 4 Geographical demographics of investigated startups

Geography of Studied Software Startups We are used to thinking of high-tech innovation and startups as generated and clustered predominantly in fertile US ecosystems, such as Silicon Valley, Seattle, and New York. However, the last decade has also witnessed the rapid growth of startup ecosystems around the world, from America and Europe to Asia. We presented below the geography of our empirical evidence, sorted by their significance (i.e., number of investigated startups) in this book (Fig. 4).

Scandinavia The Nordic countries are producing the most billion-dollar valued businesses per capita in the world, second only to Silicon Valley. While the Nordics represent only 3% of the population in Europe, it has produced more than 50% of the billion-dollar exits in the region since 2005. It is the home for industry-defining technologies and tech companies such as mobile phones, Skype, Klarna, Rovio, and Kahoot!.

The majority of cases used in this book are from Norway. We performed various approaches, i.e., survey, interview, observation, and action research, with Norwegian startups in Trondheim, Oslo, Bergen, and other places. Until 10 years ago, Norway rarely featured as a player in the European or even Nordic startup ecosystem. Fast forward to today, Norway is the fastest growing Nordic ecosystem (2016), second only to Sweden. Not all Norwegian startups have survived. In 2009, more than 40,000 new startups were founded in Norway. A year later, half of these had ceased to exist. In this particular statistic, only 27% were still operational 5 years later.

Thanks to Finnish authors, there is also a large amount of evidence in this book collected from Finland. Some of our authors have accesses to startups in the most active ecosystems in Finland, such as Oulu and Helsinki. Many startups mentioned in this book also attend Slush—the largest European events for entrepreneurs and investors. During the last 10 years, the startup movement across this country has attracted world attention, which accelerates the acquisition process of Finnish best startups. For instance, Facebook bought a Finnish startup Pryte. Japan's SoftBank paid $1.5bn for a 51% stake in Supercell—a mobile game developer after the success of Clash of Clans and Hay Day.

United States With major companies like Uber, Lyft, and Pinterest going public, the United States is the land of the most number of unicorns worldwide. Silicon Valley has dominated the US startup ecosystem for many decades. Besides, cities with popular startups include New York, Seattle, Boston, Austin, Washington DC, and Chicago. In this book, data collected from US startups are primarily from online surveys and remote interviews.

Western Europe The Italian Startup Ecosystem has been developing fast over the past few years. With many governmental initiatives and an attempt to officially register startups, Italy is working hard on increasing its support for entrepreneurs. The major geographic startup hub for Italy is Rome, with Milan being an emerging center for innovators. Several chapter authors live and work in Italian cities, making data collection straightforward.

The Dutch Startup Ecosystem has been developing rapidly over the past few years and has received great attention from both the government and the private sector, resulting in multiple initiatives to benefit young entrepreneurs. The geographic startup hubs for the Netherlands are spread around the country with Amsterdam as an innovative center. In this book, data collected from two Dutch startups are from remote interviews and online materials.

According to data from 2018, Spain is the sixth country in Europe with the highest number of digital profiles available in the employment market. The geographic startup hub for Spain is Barcelona and Madrid. In this book, data collected from two Dutch startups are from remote interviews and online materials.

Eastern Europe The Latvian startup scene is inherently quick-moving, with a fascinating hub for regional startups at Riga. One chapter of this book describes a Riga startup's journey from an early stage, scaling and failing eventually.

Russia is by far the regional leader in technological assets, number of startups, and volume of investment. Moscow and St. Petersburg appear in a list of top cities for fast-growth enterprises. Examples of successful software startups coming from Russia are Yandex and VK.ru. In this book, there is a chapter describing the story of entering the Russian market.

South America The innovation ecosystem in South America is evolving fast. The Brazilian Startup Association (Abstartups) maintains a database with the number of startups in the country (https://startupbase.com.br/stats). The picture on August 2019 when the database was accessed showed a total of 12,768 startups mapped and four main states leading the entrepreneurial ecosystem in Brazil: Sao Paulo, Minas Gerais, Rio Grande do Sul, and Rio de Janeiro.

Chile and Argentina are also two strong entrepreneurial cultures in South America, and other countries are catching up fast, such as Uruguay and Colombia. We explore some software startup research topics in such Latin American contexts.

Asia Asia increasingly sees Israel's innovativeness as a source for continued development, especially in the high-tech scene. Israel is one of the most successful startup ecosystems in many rankings. Tel Aviv's startup ecosystem has been ranked No. 3 in the world in 2018. In one chapter, we studied and compared the three startup ecosystems in Tel Aviv, San Paolo, and New York. The book also covers startup stories from other emerging Asian startup scenes, such as Singapore, Thailand, and Pakistan. Commonalities with global startups, as well as region-specific characteristics, are explored in different chapters of the book.

What Is Not in This Book? Firstly, this book is not a scientific entrepreneurship book. Even though many chapters cover topics about behaviors of startups and founders, they are not classical articles with the purpose of theory development. The chapters are practitioner-oriented, illustrated by real-world observations and case studies. Secondly, this book does not cover the financial aspects of software startups. Topics such as financial management, fundraising, cash flow management, etc. are beyond the scope of the book. Such topics would be covered in a separated specialized book. Lastly, it is noted that the content of this book highly reflects the empirical evidence we collected. The book does not address software startup situations in some developed countries like Canada, France, Germany, Belgium, etc. We also do not include cases from Australia, Africa, and the majority of Asia.

Bø, Norway — Anh Nguyen-Duc
Reutlingen, Germany — Jürgen Münch
Porto Alegre, Brazil — Rafael Prikladnicki
Bolzano, Italy — Xiaofeng Wang
Jyväskylä, Finland — Pekka Abrahamsson

References

1. Blank, S., Dorf, B.: The Startup Owner's Manual: The Step-By-Step Guide for Building a Great Company, 1st edn. K & S Ranch, Pescadero (2012)
2. Carmel: Time-to-completion in software package startups. In: 1994 Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences, vol. 4, pp. 498–507 (1994). https://doi.org/10.1109/HICSS.1994.323468
3. Creswell, J.W.: Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research, 4th edn. Pearson, London (2012)
4. Easterbrook, S., Singer, J., Storey, M.A., Damian, D.: Selecting empirical methods for software engineering research. In: Shull, F., Singer, J., Sjberg, D.I.K. (eds.) Guide to Advanced Empirical Software Engineering, pp. 285–311. Springer, London (2008)
5. Genome, S.: Global startup ecosystem report 2018. https://startupgenome.com/all-reports
6. Giardino, C., Bajwa, S.S., Wang, X., Abrahamsson, P.: Key challenges in early-stage software startups. In: Lassenius, C., Dingsyr, T., Paasivaara, M. (eds.) Agile Processes in Software Engineering and Extreme Programming, pp. 52–63. Lecture Notes in Business Information Processing, Springer International Publishing, New York (2015)

7. Kitchenham, B.A., Dyba, T., Jorgensen, M.: Evidence-based software engineering. In: Proceedings of the 26th International Conference on Software Engineering, pp. 273–281. ICSE '04, IEEE Computer Society, Washington. http://dl.acm.org/citation.cfm?id=998675.999432 (2004)
8. Klein, P.G., Bullock, J.B.: Can entrepreneurship be taught? J. Agric. Appl. Econ. **38**(2), 429–439 (2006)
9. Kuratko, D.F.: The emergence of entrepreneurship education: development, trends, and challenges. Enterp. Theory Pract. **29**(5), 577–597 (2005). https://doi.org/10.1111/j.1540-6520.2005.00099.x
10. Osterwalder, A., Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers, 1st edn. John Wiley and Sons, Hoboken (2010)
11. Perry, D.E., Porter, A.A., Votta, L.G.: Empirical Studies of Software Engineering: A Roadmap, pp. 345–355. ACM, New York. (2000). https://doi.org/10.1145/336512.336586
12. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses, 1st edn. Currency, Redfern (2011)
13. Runeson, P., Hst, M.: Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. **14**(2), 131 (2009). https://doi.org/10.1007/s10664-008-9102-8
14. Seaman, C.B.: Qualitative methods in empirical studies of software engineering. IEEE Trans. Softw. Eng. **25**(4), 557–572 (1999). https://doi.org/10.1109/32.799955
15. Steininger, D.M.: Linking information systems and entrepreneurship: a review and agenda for it-associated and digital entrepreneurship research. Inf. Syst. J. **29**(2), 363407 (2019). https://doi.org/10.1111/isj.12206
16. Sutton, S.M.: The role of process in a software start-up. IEEE Softw. **17**(4), 33–39 (2000). https://doi.org/10.1109/52.854066
17. Unterkalmsteiner, M., Abrahamsson, P., Wang, X., Nguyen-Duc, A., Shah, S., Bajwa, S.S., Baltes, G.H., Conboy, K., Cullina, E., Dennehy, D., Edison, H., Fernandez-Sanchez, C., Garbajosa, J., Gorschek, T., Klotins, E., Hokkanen, L., Kon, F., Lunesu, I., Marchesi, M., Morgan, L., Oivo, M., Selig, C., Seppnen, P., Sweetman, R., Tyrvinen, P., Ungerer, C., Yage, A.: Software startups a research agenda. E-Informatica Softw. Eng. J. **10**(1) (2016) http://www.e-informatyka.pl/attach/e-Informatica_-_Volume_10/eInformatica2016Art5.pdf

Contents

Part I
Fundamental Concepts

Six Pillars of Modern Entrepreneurial Theory and How to Use Them

Yngve Dahle, Anh Nguyen-Duc, Martin Steinert, and Kevin Reuther

Abstract In recent years, there has been an explosion of interest in entrepreneurship from both practical entrepreneurs and researchers. While theories are helpful for explaining business-driven activities in a startup, they are also valid in reasoning for the practical activities occurring in the entrepreneurial context. We believe that startups would benefit from the awareness of these entrepreneurial theories and the understanding of how they can be connected to decision-making in both business and engineering perspectives. In particular, we want to focus on theories that are already used by practical entrepreneurs and their advisors. As an example, we have studied the Scandinavian entrepreneurial ecosystem. We selected six groups of theories that might be particularly relevant for the startup population, namely (1) core competence and resource-based view, (2) effectuation, (3) the fulfillment of entrepreneurial opportunities, (4) bricolage, (5) business model innovation, and (6) lean startup. In this chapter, we explain these theories including the ongoing research around them, the connections among these theories, and how they can be applied in a real case study.

Keywords Entrepreneurship theories · Effectuation · Bricolage · Resource-based view · Lean startup · Business modeling · Software startups · Software development

Y. Dahle (✉) · M. Steinert
Faculty of Engineering Science and Technology, Norwegian University of Science and Technology, Trondheim, Norway
e-mail: yngve.dahle@ntnu.no; martin.steinert@ntnu.no

A. Nguyen-Duc
Department of Business and IT, University of South-Eastern Norway, Notodden, Norway
e-mail: angu@usn.no

K. Reuther
School of Business and Enterprise, University of the West of Scotland, Glasgow, UK
e-mail: kevin.reuther@uws.ac.uk

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_1

1 Introduction to the Theoretical Map

1.1 The Need for Entrepreneurial Theories

In the small Scandinavian country of Denmark, there are 216 different public organizations set up to give advice and support to entrepreneurs, managed by 110 different branches of the government. The setup in Norway and Sweden is very similar. The Danish "*Simplification Committee for Business Promotion*" released its annual report [1] in April 2018, suggesting a new policy to manage the public entrepreneurship support. The initiative shall help to ensure that the business in the regions and the nation will be globally competitive in the future. Four main areas for improvement in Denmark include (1) creating a coordinated entrepreneurial journey, (2) being able to give high-quality advice independent of the individual experience of each advisor, (3) using modern technology to create an effective knowledge transfer, and (4) being able to learn from the collected data. This is the start of a process aiming to build a national best practice for entrepreneurship in order to give entrepreneurs optimal support—and to build a structure for gathering behavioral data to continuously improve this process. Similar initiatives are being initiated in Norway and Sweden. Strangely enough, the academic institutions within entrepreneurship do not seem to be very much involved in this development. One might argue that this separation into a "*consultant-driven*" practical and an academic branch of entrepreneurial thinking seems to be partially factual in all of the three Scandinavian countries.

Entrepreneurship research is relatively new as an academic field, but it has a long tradition [2]. The term "*entrepreneur*" has been used in the French language since the twelfth century, describing people who trade within or among European cities, buying goods cheap and selling them at a higher price [3]. The first economist to focus on the role of entrepreneurship in economic development was Joseph A. Schumpeter [4, 5]. In his seminal work, Schumpeter tried to develop a new economic theory based on change [5], arguing that "*creative destruction is the essential fact about capitalism*" [4]. The "*twin oil crises*" in the 1970s triggered a reappraisal of the role of small firms. Many large companies were hit by severe economic difficulties. The increased interest in smaller firms can be attributed to (1) a fundamental change in the world economy, related to the intensification of global competition, the resulting increase in the degree of uncertainty, and greater market fragmentation, and (2) changes in the characteristics of technological progress giving large firms less of an advantage [6]. The new trends are also reflected in numerous scholarly works concerning entrepreneurship and the role of small business. The explosion in the number of entrepreneurship-oriented journals in the 1980s and 1990s reflects the similarly dramatic increase in entrepreneurial activity that took place at the same time [7, 8]. Still, the vast amount of knowledge created in this research does not seem to be fully utilized by practical entrepreneurs and incubators, grant providers, and advisors who are given the task to support them.

The common topics

- ✓ Entrepreneurial process
- ✓ The existence, discovery, and exploitation of entrepreneurial opportunities
- ✓ New venture formation
- ✓ The social and environmental context
- ✓ New venture finance
- ✓ Macroeconomic impact of entrepreneurial activity
- ✓ Characteristics of entrepreneurs

One reason for the chasm between the practical and the academic branch of entrepreneurship may be the latter's lacking ability to translate knowledge of entrepreneurship into practical, normative advice for the entrepreneurs and the organizations that help them. In 2012, Scott Shane wrote an article [9] reflecting on and summarizing the classic 2000 article "*The promise of entrepreneurship as a field of research*" [10] that he co-wrote with Venkataraman. In this article, he addressed the major challenge of normatively trying to find a "*best-practice*" of entrepreneurship:

> We did not intend to say that the entrepreneurial process is rational, planned, strategic, or even temporarily ordered, but merely that the entrepreneurial process has sub-processes. There may be no optimal entrepreneurial process, allowing for many equally effective approaches, which is an important issue for the field to explore. It is also possible that one approach may be optimal but that many entrepreneurs do not approach the process "the best way". This point has important ramifications for the fields desire to be normative.

Such a normative model would address the practical problems that must be solved to actually improve the success rate of entrepreneurs. Why do some startups succeed while others fail? What is the essence of entrepreneurship? Who is most likely to become a successful entrepreneur and why? How do entrepreneurs make decisions? What environmental factors foster the most successful entrepreneurial activities? Entrepreneurship research is plagued by these and other unanswered fundamental questions. In the early days of the field, it has been much criticized for a lack of cohesive explanatory, predictive, or normative frameworks that are able to explain such questions [10, 11]. The purpose of this chapter is to try to use modern entrepreneurial theory in such a way that it can bridge the gap between entrepreneurship researchers on one side and practical entrepreneurs and their supporters on the other.

1.2 Selections of the Theoretical Blocks

Scholars have proposed several theories to explain the field of entrepreneurship. These theories have been categorized into six different sub-domains, all drawing from different fields: (1) Economic entrepreneurship theory, (2) Psychological entrepreneurship theory, (3) Sociological entrepreneurship theory, (4) Anthropolog-

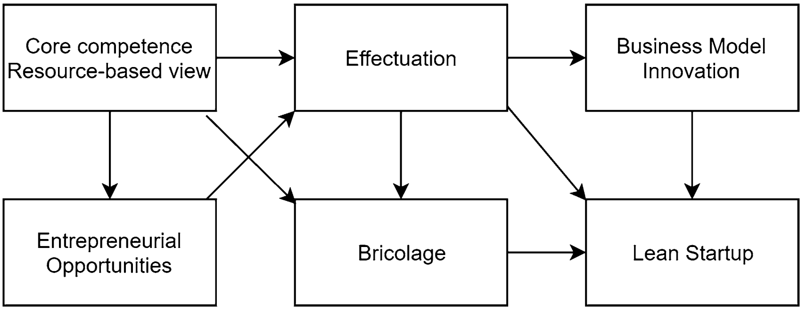

Fig. 1 A theoretical map of entrepreneurship literature

ical entrepreneurship theory, (5) Opportunity-based entrepreneurship theory, and (6) Resource-based entrepreneurship theory [12]. Targeting the intersection between entrepreneurship and software product engineering, we look for theories that can be applied in technology-relevant contexts, such as explaining technical decisions under business constraints, like deciding on a set of product features basing on the current resources and visions. Therefore, we limit ourselves to theories at team or business levels, focusing on the managerial, process, and environmental factors of entrepreneurship. Hence, we have excluded economic-based theories (entrepreneurship as an economic system), sociology-based theories (entrepreneurship as a social system), anthropology-based theories (culture, communities of entrepreneurs), and psychology-based theories (personal characteristics of entrepreneurs). Our ambition is not to present a complete picture of entrepreneurial theory, but rather to focus on theories that are usable for, and actually used by entrepreneurs and their helpers. With this perspective, we select six theories (as shown in Fig. 1) as instrumental blocks for the modern, process-based, and practical approach to entrepreneurial studies. They reflect the opportunity-based and resource-based views on entrepreneurial activities. The six pillars seem to be closely related to each other as they are sharing many of the same ideas and concepts. We will try to describe these relationships in the first part of this chapter. The chosen theories also have the advantage that they are relatively well known to practical entrepreneurs, advisors, and mentors in incubators and entrepreneurship support programs. This may help to increase the adaptation of the model.

We are starting with the concept of the Resource-Based View (RBV) [13, 14], with a special focus on Core Competence as a resource [15]. We then look at entrepreneurship as a nexus between entrepreneur and opportunity [10] and try to build a bridge via the discussions around Effectuation [16, 17], Bricolage [18], and Business Model Innovation [19–22] toward the Lean Startup Movement [19, 23, 24]. These six building blocks build on each other, to a varying degree. The model starts with the earliest contribution in the top left corner and shows the influences the pillars have had on each other toward the bottom right corner. The detailed relations will be explained in the following sections.

We choose Shane and Venkataraman's definition of entrepreneurship [10]: "*the scholarly examination of how, by whom, and with what effects opportunities to create future goods and services are discovered, evaluated, and exploited*." This process-based definition has got some great advantages for the purpose of our research. Firstly, it follows Gartner's example of focusing on what the entrepreneur *does* rather than on trying to analyze what the entrepreneur is, or on the personal traits of the entrepreneur [25]. Trying to understand the entrepreneur based on his/her activities is the core of the proposed model to be developed in the course of this chapter. Secondly, the definition does not put great emphasis on the establishment of companies. A significant share of entrepreneurial activities often takes place before an actual company is established. Similarly, entrepreneurial activities often take place as improvements of processes in well-established companies. This is sometimes referred to as "*Intrapreneurship*" [26]. Thirdly, the focus on the nexus between the entrepreneur and the opportunity seems to be the starting point for the incremental and effectual approaches utilized by the Lean Startup Movement.

2 The Illustrative Case: QuickNews

In order to illustrate the practical usages of the six theoretical pillars, we have conducted a single case study. The case is not necessarily representative for successful startups, but it is a typical digital startup facing engineering and business decisions from which we can acquire various insights. *QuickNews* is a spin-off from a Norwegian social media company. The CEO of the company quit her job and sought for a technical team to develop a hyper-local news platform. The business idea came to her mind in the early days of 2014. She started with the business idea and hired several consultants, freelancers, and contractors to realize and refine the idea. The CEO explored her networks with local incubators, investors, a university, and other startups to seek for technical competence, funding, and business support. A CTO who used to be an IT researcher joined the team and initiated a prototyping contract with a foreign outsourcing team. The outsourcing team worked in a Sprint-based approach adopting Sprint planning and retrospective meetings, burn down chart, and communication via social media. The first Minimum Viable Product (MVP) was created in late 2015. The product was launched at the end of 2015 (Table 1).

During the journey of QuickNews, several questions about both business and product development had been asked and decided by the founders. The example questions are:

- Which feature should be built first?
- Whom should we involve in the development/marketing campaign?
- How do we generate benefits from the current business model?
- Who should be the partner in the prototyping phase?
- How can the software be created to attract target users?

Table 1 Case demographics

Startup duration	April 2015
Team size	2 (early stage)–6 (startup stage)
Team competence	CEO (journalism, marketing), CTO (Information technology), 01 content developer, 03 software developers
Product idea	A hyper-local news platform where citizens are journalists
Funding	Seed funding from the government (2 rounds), angel investor (1 round)—total budget ca 500,000 euros
Traffic	200+ users, 1500+ posts
Core features	Post a news article, view a news article in a list view and a map view, livestream video, news recommendation, award system for posts
Market	Norway, Singapore

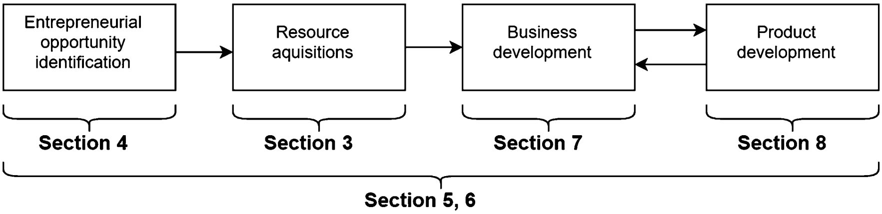

Fig. 2 Mapping the case examples to chapter sections

- How can the desired features be built within a given budget?
- When should we launch the product?
- Which marketing channels/content should we focus on at the early stage?
- Where do we find lead users?
- What is the best way to validate the market demand for the product?

The study of QuickNews was planned when the second author participated in the case as a co-founder. The purpose was to (1) become a successful co-founder and (2) understand and research about startups. At that time, startups were an unfamiliar phenomenon for explorative investigation. The study was designed as a type of ethnographic studies [27] based on participant observations [28] including documentation of field notes, retrospective meetings, and workshops. The second author participated at and observed various working sessions and meetings in QuickNews, getting insights on what, when, why, and how such questions are raised and addressed. The study was conducted from September 2015 to May 2017.

In the following sections of this chapter, we use entrepreneurial theories to explain some of the scenarios occurring in QuickNews. As illustrated in Fig. 2, we explain how the business idea was initiated, what technical competence was added to the project, how the business and product coevolved, and what consequences the restricted resources in early-stage startups had for QuickNews.

3 Core Competence and the Resource-Based View

The concepts of "*Core Competence*" [15] and "*Resource Based View*" [13] have been central building blocks for the process-based entrepreneurship thinking developed in our millennium. J. Barney [13] challenged "The *Environmental Models of Competitive Advantage*," which tries to understand competitive advantages by primarily analyzing the organization's external opportunities and threats. This latter model draws heavily on Porter's Five Force Model [29], where the company is understood in accordance to a competitive ecosystem containing threats from new entrants and substitutes, the bargaining power of suppliers and customers and finally, the existing competitors within the industry. As an alternative, Barney has introduced the "*Resource Based Model*," which examines the relationship between firms' internal weaknesses and strengths and its performance. These resources, according to Barney, consist among other things of the firms "*assets, capabilities, organizational processes, firm attributes, information and knowledge*." Wernerfelt has defined resources as "*anything that could be thought of as a strength and a weakness of a given firm. More formally, a firm's resources at a given time could be defined as those (tangible and intangible) assets that are tied semi-permanently to the firm*" [14]. Barney developed the VRIO model structured in a series of four questions to be asked about the business activities a firm engages in: (1) the question of Values; (2) the question of Rarity; (3) the question of Imitability; and (4) the question of Organization [13]. The answers to these questions determine whether a particular resource or capability for the firm is a strength or a weakness. The VRIO model describes ways that firms can expect to be successful. The RBV is closely related to the Dynamic Capability View [30].

Prahalad and Hamel extended the RBV in their discussion of the "*Competence Based View*" [15]. Note that there is a key distinction between resources and competence. The individual resources of a startup include intellectual assets, patents, hardware, software infrastructures, and brand names. Competence is the capacity of a team of resources to perform some task or activity. This theory enhances the Resource-Based Model by focusing further on the Resources that constitutes "*Unique Knowledge*" in the organization. They claim that the competitive advantage of a firm is better understood by researching the core competencies behind products than researching the products itself. While resources are the source of the firm's competencies, competencies are the main source of its competitive advantage.

The implications

- ✓ Highlights the necessity of firms to develop superior internal resource and competence of all their members.
- ✓ Assesses existing resources and competences as strength or weaknesses.
- ✓ Invests on better resources and consequently superior capabilities as a way of reaching higher levels of growth.

Table 2 Assessment of AI competence in the team

Element	Explanation	Evaluation
Value	The resource enables the company to implement their strategies	Yes: the product will be able to recommend news according to personal preference
Rarity	The resource can only be acquired by one or few companies	Yes: the capability of implementing previous features
Imitability	The resource should be hard and costly to imitate or substitute	Yes: without the recommendation feature, the product is much less competitive in the market
Organization	The company is organized to adequately exploit these resources	No: the new resource might not fit into the current backend development team

From the project management perspective, entrepreneurs can analyze and manage their limited resources in the early stages of the project life cycle. In QuickNews, after 6 months of recruiting the CTO, the company comes up with the question if they need to hire a new Artificial Intelligence (AI) scientist. The CEO would like to have a feature that recommends news based on personal preference but is not sure about the impact of the action. By using the VRIO model as a postmortem analysis, she can assess the value of hiring an AI scientist for QuickNews (Table 2).

4 Entrepreneurship as the Fulfillment of Opportunities

The concept of an "entrepreneurial opportunity" is central for the study and theory of entrepreneurship. Shane and Venkataraman defined entrepreneurial opportunities as "*situations in which new goods, services, raw materials, markets and organizing methods can be introduced through the formation of new means, ends, or means-ends relationships*" [10]. Entrepreneurs, individuals who pursue entrepreneurial opportunities, must believe that the value of resources used according to a particular means-ends framework would be higher than if exploited in their current form. Without entrepreneurial opportunities, there will be no "*entrepreneurship*." The definition is extended into describing the focus of entrepreneurship research into answering three questions:

- Why, when, and how opportunities for the creation of goods and services come into existence?
- Why, when, and how some people and not others discover and exploit these opportunities?
- Why, when, and how different modes of action are used to exploit entrepreneurial opportunities?

Shane later expanded on the difference between the objective opportunities and the subjective recognition of those opportunities [31]. He separates between entrepreneurial opportunities and business ideas by stating that the former are situations where resources can be combined to generate profit, while the latter is the entrepreneur's interpretation of how these resources should best be used in pursuit of the opportunity. Jonathan et al. discussed three types of opportunities: (1) By the locus of the changes that generate the opportunity; (2) By the source of the opportunities themselves; and (3) By the initiator of the change [32]. According to him, entrepreneurial opportunities can, in fact, occur because of changes in a variety of parts of the value chain. Certainly, the creation of a new good or service can create an opportunity for entrepreneurial profit, as is the case when the development of an accounting software or a surgical device makes possible a product or service that can be sold for more than its cost of production. Finally, new methods of production, such as the assembly line or computer-aided drug discovery, have provided opportunities for entrepreneurial profit.

Opportunities can also vary as to their source. Eckhardt et al. suggested four major categories of opportunity sources [31]. The first type comes from asymmetries in existing information between market participants. Changes in technology, regulation, and other factors generate new information about how resources might be used differently. This information changes the price for the resources, hence allowing companies who have early access to the new information to purchase resources at low prices, use the information to create products or services, and sell them at an entrepreneurial profit.

The second type of opportunities comes from exogenous shocks of new information. Several types of exogenous shifts exist, for example, government policies, social and demographic changes, and those generated by the creation of new knowledge.

The third source of opportunities comes from the trade-off between the supply and demand sides. Opportunities can also be classified on whether the changes that generate them exist on the demand or the supply side. Customer preferences influence the allocation of resources because producers need to respond to the preferences and purchasing habits of consumers. Thus, demands change from exogenous shifts in culture, perception, tastes, or mood can open up opportunities.

The fourth type is from attempts to increase productivity in organizations. The merger or breakup of organizations can create productive opportunities as new customer relationships or economies of scale are generated.

Opportunities can also be different due to their initiators of changes. Different types of entities initiate the changes that result in entrepreneurial opportunities, such as specialized knowledge creating agencies, such as universities or research laboratories that lie outside the industrial chain, and firms within the industrial chain, including suppliers and customers.

Considering the concept of opportunities, we choose a quite wide interpretation of the term in one specific regard. The textbook understanding of an opportunity is a rather positive one. You will have the opportunity to go from a good situation to increase your financial and human utility. You decide to quit your job to build up a company—and the purpose will be to create a growing entrepreneurial organization by creating value for the involved stakeholders. This will be your typical "*opportunity*" entrepreneur. In many areas of the world, and in many vertical industries, however, entrepreneurship is motivated by the lack of other meaningful employment alternatives. In developing economies, traditional employment opportunities are few and far between, and we also see a large number of "*art entrepreneurs*" emerging [33]. These are "*micro-entrepreneurial*" projects started by people who have no other ambition than to make a living for themselves from their talents. These will constitute what we call "*entrepreneurs out of necessity*." Being self-employed is a necessity coming from the situation they find themselves in Venkataraman and Shane do not specifically separate between these two categories, and we will treat "necessity" as a form of opportunity.

Treating entrepreneurship as a separate science, a clear and agreed upon definition and the focus on understanding the nexus between the entrepreneur and the opportunity should make it easier to build the practical advice to entrepreneurs upon existing theory. This has created a platform for both Effectuation [16] and Bricolage [18]. In addition, it has created the foundation for the focus on understanding entrepreneurial projects by understanding their Business Models, and by The Lean Startup movement—who build their understanding of entrepreneurship on a set of iterative improvement in the recognition and exploitation of opportunities [24].

Locus of the change

- ✓ The creation of new products or services
- ✓ Discovery of new geographical market
- ✓ Discovery of new raw materials
- ✓ New methods of production
- ✓ New ways of organizing

Source of opportunities

- ✓ Resource information asymmetry
- ✓ Exogenuous shift of social, institutional and technical conditions
- ✓ Change in the balance of demand–supply
- ✓ Merge activities
- ✓ Initiators of changes
- ✓ Specialized knowledge creating agencies
- ✓ Industrial supply chain

Capturing different types of opportunities and the journey of opportunity identification makes entrepreneurs aware of opportunities that they might have neglected earlier. This can trigger entrepreneurial activities. Figure 3 illustrates the process of realizing an entrepreneurial opportunity in QuickNews.

The initial entrepreneurial opportunity in QuickNews arises from the CEO's ability to accumulate her social capital and existing experience and from that reach the level of entrepreneurial alertness leading to forming the entrepreneurial team. Here we can see the close link between the focus on resources in the previous chapter, and the focus on finding and exploiting opportunities here. The initial entrepreneurial opportunity arises directly from the CEO's knowledge and

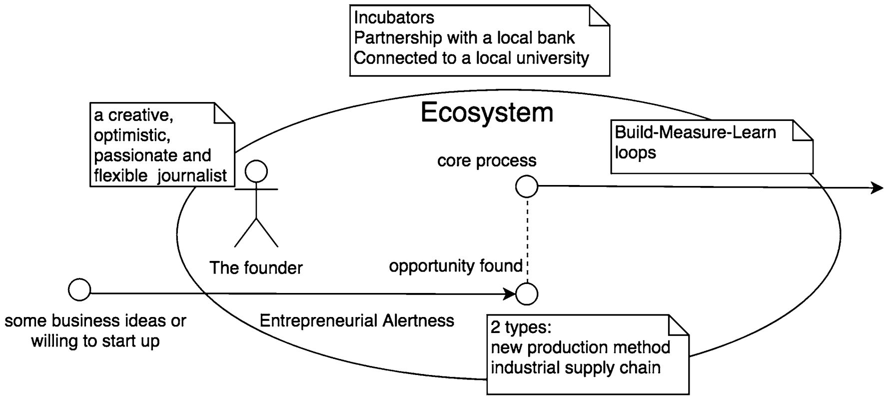

Fig. 3 Entrepreneurial opportunity identification process of QuickNews

network resources—more than from a specific market event. The second type of entrepreneurial opportunity, which is a change in the industrial supply chain, is discovered later during the business development.

5 Effectuation

Sarasvathy adds to the understanding of the opportunity-seeking entrepreneur with her theory of Effectuation [16]. She claims that an entrepreneurs' journey starts up without clearly defined objectives. An entrepreneur typically starts with only some very general goals "*such as the desire to make lots of money or to create a valuable legacy like a lasting institution, or, more common, to simply pursue an interesting idea that seems worth pursuing*." Because of these characteristics of entrepreneurship, Sarasvathy argues that an effectuation approach gives much more meaning than a causation approach when studying it [16]. She defines the difference between the two as follows:

- Causation processes take a particular effect as given and focus on selecting between means to create that effect.
- Effectuation processes take a set of means as given and focus on selecting between possible effects that can be created with that set of means.

Sarasvathy uses an example where a chef is cooking dinner [16]. In the Causation scenario, he gets a menu from a client, gets the ingredients, and cooks the meal according to the menu. The start is the menu, and the process is to select how to prepare the meal. In the Effectuation scenario, the client asks the chef to cook the meal from the ingredients he can find in the kitchen. Here, the chef has to create the meal from a given set of resources—thus being in a much more creative than analytical process.

This means that the entrepreneur starts with the resources he has at hand and tries to create as much value as possible from that starting point, rather than having a clear target destination and trying to find the necessary resources to get there. In this way, this theory is related to the "*Resource Based Model*" [13, 14, 34]. Thus, entrepreneurship is seen as a creative effort rather than an analytical one, and as a much more experimental and design-driven process than traditional strategic management. This also is part of the foundation for the Lean Startup Movement [24].

Sarasvathy also introduces five principles of entrepreneurship, presented as "*five habits of highly effective entrepreneurs*." These are described as: (1) Bird in hand: You should start with what you have, asking questions like "What are my characteristics and preferences?," "*What can I do?*," and "*Who do I know?*." This draws on the RBV and we will see that it is highly linked to the concept of bricolage. (2) Affordable loss: You do not invest more into the entrepreneurial project than you can afford to lose. (3) Crazy quilt: You should seek alliances with potential actors in your entrepreneurial ecosystem. (4) Lemonade: If you suffer a setback, you should turn it into an opportunity. This is the same iterative logic you will find in the Lean Startup thinking. (5) Pilot in the plane: You should take charge of the situation, using the other four principles to make things happen.

The logic of effectuation also builds on the "Science of the Artificial" by Simon [35], who clearly sees sciences like entrepreneurship from a similar design viewpoint: "*Finally, I thought I began to see in the problem of artificiality an explanation of the difficulty that has been experienced in filling engineering and other professions with empirical and theoretical substance distinct from the substance of their supporting sciences. Engineering, medicine, business, architecture, and painting are concerned not with the necessary but with the contingent—not with how things are but with how they might be—in short, with design*."

Effectuation principles

✓ Bird in hand: Start with who you are, what you know
✓ Affordable loss: Invest only what you can afford to loose
✓ Lemonade: Be open to surprises and leverage them
✓ Crazy quilt: Build stakeholder partnerships
✓ Pilot in the plane: Focus on action where you directly influence the outcome

Effectuation is considered an effective framework to explain for various engineering activities occurred in early-stage digital startups [36]. It is important for technical co-founders that technical decisions are not only influenced by the technologies and engineering processes, but also effectual situations of the business. As a postmortem analysis, we used Sarasvathy's five principles to explain for the challenges in the prototyping process in QuickNews, as shown in Table 3:

✓ Bird in hand: QuickNews starts with a few wireframes, websites, and Facebook page done by a consultant when the CTO had not been onboard. After the CTO was onboard, they built a client–server mobile application and evolve from it.

Table 3 Adoption of Build–Measure–Learn loop

Hypotheses	Actions	Descriptions
People are interested in trusted validated news	Build	MVP1: a Facebook landing page describing the business idea, a survey on pain points of the existing news platforms. The MVP was done in 2 weeks.
	Measure	Survey feedback on the satisfaction with the new features.
	Learn	Hypothesis validated. News and photos should not be uploaded but generated right in the field.
People are willing to share hyper-local news around them	Build	MVP2: a single feature: post a news and view the news in a map-view. The MVP was done in 3 weeks.
	Measure	Amount of posts, types of posts in the apps.
	Learn	Hypothesis validated. A camera-ready button helps user to engage in information sharing via taking photos.
People are interested in sharing news via an interesting sharing mechanism	Build	MVP3: an evolutionary prototype with the new feature of live capturing photos and simple text tagging.
	Measure	Qualitative feedbacks for test users.
	Learn	The camera button works, but it should have an option to disable it due to energy consumption.

The CEO has stated "*Without Mr. Z, we would not be able to have the functional MVP as we have today. He and his team have a competence of doing this*."

✓ Affordable loss: The CEO started the company with her thoughts "*I have always wanted to develop this idea. I think I can afford to take two years and invest X. NOK to try this out. In the worst scenario, I will lose the money and I will be back to the job market in two years*." The CEO accepted a certain amount of risk as inevitable in all situations.

✓ Lemonade: QuickNews had developed a feature of taking photos for a post. It was a difficult task as the customer (or early adopter) was not satisfied with the developed feature. The team came up with an idea that the camera was put in a holding situation when the post is initiated. This had led to the idea that the product needs a feature of capturing a short video clip (this is done by putting the camera into recording state instead of holding state). The feature is now one of the core features of the platform.

✓ Crazy quilt: The journey of QuickNews is the path of building partnerships rather than beating competitors. The CEO often takes her MVPs to the nearest potential customer. Some of the people she interacts with making a commitment to the venture, committing time and/or money and/or resources and, thus, self-select into the new venture creation process. This was the way the CTO has joined QuickNews and became a co-founder. This was also the way that *Addresseavisen* (a local newspaper) became a partner and helped QuickNews with organizing events. This was also the way that Makerfaire (a local event organizer) became a partner by adopting the platform to make reports from the events they arranged.

✓ Pilot in the plane: In QuickNews, the CEO pursued the work with things within her control and people who want to help co-create it instead of attempting to predict the future, determining the perfect timing, or finding the optimal opportunity.

6 Bricolage

Another viewpoint on entrepreneurship is "*Bricolage*" [18]. Traditional economic theory is based on an assumption of perfect financial rationality. This means that if an entrepreneurial case is able to argue an acceptable return on investment, the necessary funding to buy all the resources needed should be available in the financial market.

An alternative understanding of entrepreneurship is starting with the fact that for different reasons, most entrepreneurial projects are starting under an extreme scarcity of resources. They may not be able to get the necessary funding, or they may be unwilling to give away the control over the company that would be the consequence of such funding. Therefore, they manage without funding and utilize creativity and practical sense to reach their objectives rather than the strict rationalism of traditional economic theories.

The theory of "*entrepreneurial bricolage*" aims at explaining what entrepreneurs do when facing such resource constraints. Bricolage is defined as "*making do by applying combinations of the resources at hand to new problems and opportunities*" [18].

Their definition builds on three concepts from Levi-Strauss [37]. The first is: "*Making do*," where the entrepreneurs "always have to make do with whatever is at hand." This seems to be in line with the RBV. If utilizing existing resources is vital for entrepreneurial success, having a good understanding of what these resources are must be vital. Furthermore, the "*combination and reuse of these resources for new applications*" is an important part of bricolage. Finally, Levi-Strauss described "*the resources at hand*," meaning that entrepreneurs gather physical artifacts, skills, or ideas on the principle that "*they may always come in handy*," rather than as a part of a predefined plan [20]. The idea of this "*artistic improvisation*" seems to be clearly related to the thinking behind effectuation. This is also related to the iterative developments in the Lean Startup theory.

The entrepreneurs who engaged in bricolage tended to maintain a broad collection of tools, parts, and other physical resources for which no immediate need was available. Over time, they found use for these resources. With respect to customer needs and the market, the bricoleurs tended to serve anyone they could instead of focusing on certain types of customers. Instead of maintaining a professional distance to their customers, many bricoleurs had an informal approach to customers, in which the entrepreneurs and relevant stakeholders often formed a community of friends. Baker and Nelsons proposed that applying bricolage in limited areas ("*selective bricolage*") may enable firms to grow, whereas excessive ("*parallel*") bricolage may lead to the opposite outcome.

Bricolage principles
- ✓ Making do with whatever is at hand
- ✓ Combining and reusing these resources for new applications
- ✓ Gathering the resources at hand, as they may always come in handy

When the CEO of QuickNews started her company, she had to rely on limited funding from the Norwegian state and an angel investor. Therefore, she had to make do with the funding she had and started her project utilizing the existing people in her network. For validating an interest of the local community on certain types of news, the CEO needed a platform where people could post news and be notified. Due to the limited budget and technical competence, she made use of the news platform that she had access to in her old company. She also convinced a technical person in her old company to help her to operate the platform. When she was able to secure sufficient funding, she hired a development team to build her own platform. Such a way of "*making do*" with the available resources shows signs of Bricoleur behavior.

7 Business Model Innovation

Practical entrepreneurs in Scandinavia have been using different Business Model Canvases (in particular the BMC) as a tool to develop their projects for quite some time, and these canvases are the core of the communication between the entrepreneurs and their different advisors. We think it is valuable to note that the concept of Business Model Innovation goes further than utilizing Business Model Canvases to analyze individual entrepreneurial projects. The whole process of analyzing different projects, categorizing activities from these projects into different element archetypes, and then using these archetypes to give better advice has been a great contribution to the improvement of entrepreneurial projects.

Ritter and Lettl have done an extensive review of business model literature, identifying five different perspectives on the term Business Modelling [38]: (1) *business-model activities* is a description of the activities necessary to fulfill the entrepreneur's strategy; (2) *business-model logic* is a description of the value-creation logic created by these activities; (3) *business-model archetypes* are examples of such logic, which are well known and proven, like the freemium–premium models or recurring revenue models; (4) *business-model elements* describe the business model logic as a subset of elements, like unique value proposition and customer segments; and finally, (5) *business-model alignment* describes the way the different elements interact with each other.

So, *Activities* build value-creation *Logic*. These Activities can be categorized into different *Elements* that must *Align* with each other. Some of these combinations have been known as *Archetypes*.

Osterwalder et al. have a similar approach [20, 21]. They start with a definition: "*A business model is a conceptual tool containing a set of objects, concepts and their relationships with the objective to express the business logic of a specific firm. Therefore, we must consider which concepts and relationships allow a simplified description and representation of what value is provided to customers, how this is done and with which financial consequences*" [21]. Next, they describe a set of business model elements in what they call a "*meta-model*." They then use these elements to separate different taxonomies of types, which they compare to real business models of real companies.

Zott et al. [22] introduce four core points that describe business models as: "(1) a new unit of analysis that is distinct from the product, firm, industry, or network (2) a system-level, holistic approach to explaining how firms "do business"; (3) conceptualizations where activities play an important role, and (4) an explanation of both value creation and value capture."

Here it might again be interesting to focus on one individual word in the term. The word model is defined as "*a simplified description and representation of a complex entity or process*" [21]. One could think of an architect's cardboard and Styrofoam model (or computer simulation in 2019) of a building or a shipbuilder's small-scale model of a ship. The purpose of this is the ability to simulate changes and improvements before needing to undertake the investment of building the object in full scale. This seems consistent with how businesses actually use the business model today. A business model is often used as a theoretical simulation of what the entrepreneur wants to do with her business [39, 40]. The results from these simulations are used to change the real business parameters. This means that the business model can be used both to describe the business to external stakeholders, and actually to change and improve it.

Around the change of the millennia, a number of such element-based platforms were released, and over the next years, a set of second-generation platforms came. To provide some examples, we will describe the elements of three different platforms from 2005, 2010, and 2012, respectively. As a part of this, we are going to discuss the logic of introducing the business idea or business mission to separate between the External (customer-centric) and the Internal (entrepreneur-centric) business model.

Morris' Business Model setup consists of six elements [41]: How will the firm create value? For whom will the firm create value? What is the firm's internal source of advantage? How will the firm position itself in the marketplace? How will the firm make money? What are the entrepreneur's time, scope, and size ambitions?

Osterwalder and Pigneur's Business Model Canvas [21] is the most widely used version of a Business Model structure. The canvas has nine segments or elements. The first is the *Customer Segment*, in which you describe who your company creates value. Then comes the *Value Proposition*, in which you describe what value the solution brings to the customer. The *Key Channels* are describing how you contact customers and how you reach them. *Customer Relations* describe which type of relationship you have through your channels. *Revenue Streams* describe how you are paid. *Key Resources* are addressing what type of resources the company needs to create value for the customer. *Key Activities* describe the activities you must yourself carry out to fulfill the value proposition. *Key Partners* are allies who help to make the value proposition possible. Finally, *Cost Structures* address the most significant cost drivers in order to operate the business model.

The Lean Canvas by Ash Maurya has simplified the BMC [19]. His version is different from Osterwalder and Pigneur's in four ways; key partners are replaced by problems, key activity with solutions, customer relations is with unfair advantage, and key resources with key metrics.

Both models moved the general concept of strategy work in a new direction. From a "*Lean Startup*" point of view, the advantage compared to traditional strategy models, is the ability to absorb changes coming from the surroundings. If the customer problem changes, the value proposition should change. Both canvases allow you to see this, and immediately change all the other boxes to fit the new value proposition.

Finally, we want to argue that there is a strong relationship between business models and business ideas or business missions [42]. The business mission consists of the answer to three questions:

- Key Market: "Who are your customers and target group?"
- Key Contribution: "What do you do for this customer?"
- Distinction: "Why does the customer choose you?"

This creates a logical separation between the internal and the external elements of the business model. Whereas the external business model should see the world from the customers' viewpoint and describe *what* the entrepreneur should be doing, the internal business model (the traditional business model) describes *how* the entrepreneur should organize his/her activities. Using both these terms makes it easier to separate between the problem you should solve and the solution to this problem.

Business model innovati on

✓ Business modelling is a process deriving normative models of entrepreneurship from experience.

✓ Three famous represent-actions of a business model are the Morris model, the Business Model Canvas, and the Lean Canvas.

The BMC model had been selected to capture the business model at the beginning of QuickNews is shown as in Fig. 4. We see how this is an integrated description of the relevant business elements of QuickNews. Each element of the canvas has also evolved over time, as the team has gained experience with what works and what does not work.

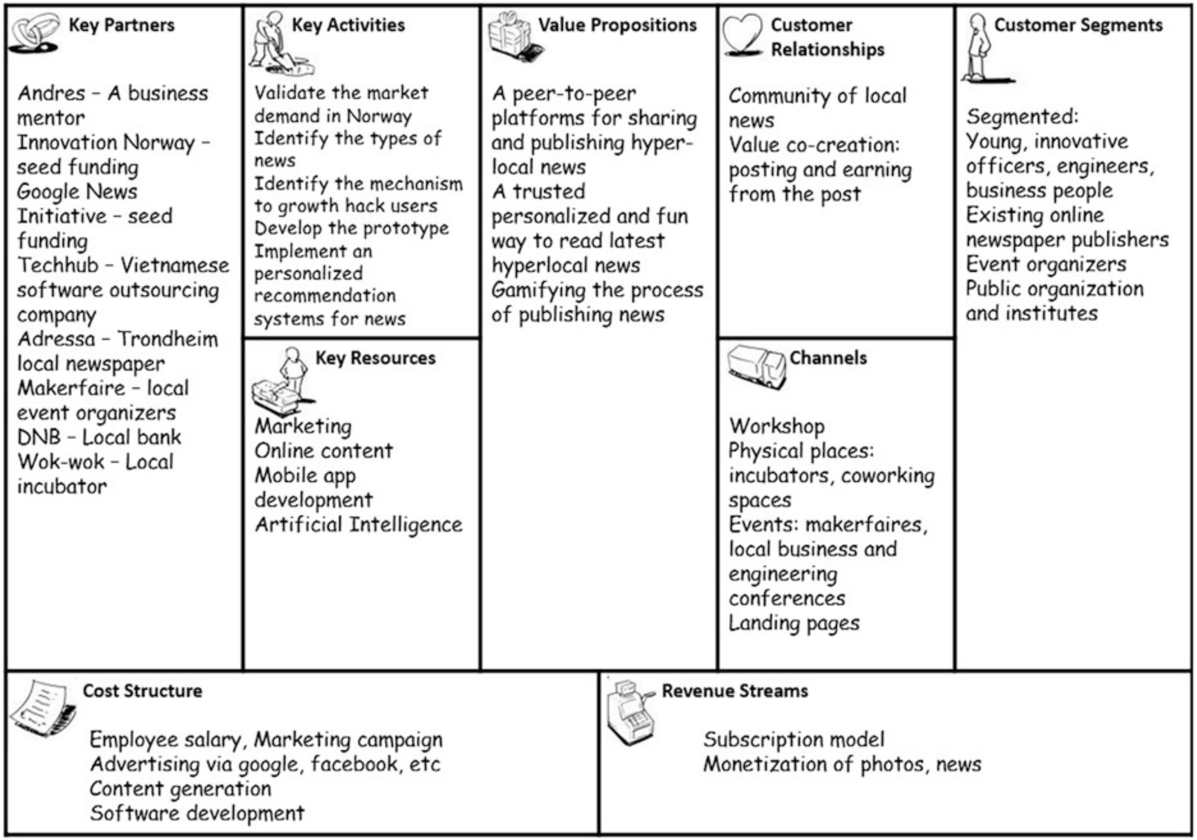

Fig. 4 Business model canvas of QuickNews

8 The Lean Startup Movement

The Lean Startup Movement [24] draws heavily on Shane and Venkataraman [10]. But, instead of looking at *one* nexus between the entrepreneur and the opportunity, their approach is that an entrepreneurial process is a constantly changing dynamic between opportunity and entrepreneur. The three questions:

- Why, when, and how opportunities for the creation of goods and services come into existence?
- Why, when, and how some people and not others discover and exploit these opportunities?
- Why, when, and how different modes of action are used to exploit entrepreneurial opportunities?

are central in the Lean Startup methodology. The only way the use differs is that the three questions can be seen as a systematically iterative process where the questions are asked over and over again, while the answers are being constantly improved upon. The Lean Startup theory is that it is difficult to make long-term plans and objectives given the extreme uncertainty in entrepreneurship. Succeeding with entrepreneurship can be done by treating every entrepreneurial project as an ongoing experiment while utilizing as few resources as possible. This means working closely with customers [23], creating small and inexpensive experiments [24], and then trying to "fail fast" by trying out the riskiest actions first [19].

This iterative development builds on existing resources, in particular on the core competence of the business. The approach is rooted in Effectuation, as it is describing processes where the objectives are being developed as the projects take shape, and where the method is explorative and creative rather than analytical. It also seems to be covering all the three of Levi-Strauss concepts of Bricolage [38]. A Lean Startup company most often has to make do with "*whatever is at hand*." They constantly have to "*recombine and reuse resources for new applications*" and "*gathers resources that may come in handy, rather than as a part of a pre-defined plan*." The Lean Startup movements also draw heavily on Business Modelling, where in particular the Business Model Canvas is an integral part of the model. This link between Bricolage, Effectuation, and The Lean Startup Movement is also described by Ghezzi: "*With regards to the logic behind entrepreneurial behavior, the entrepreneurs showed how adopting and implementing LSAs (Lean Startup Approaches) drove them to take an "effectual" or "bricolage" stance on many occasions, underscoring how these logic are, in essence, connected to LSAs' main steps and constituting elements*" [43].

Another approach and toolkit that are gaining increased traction and widespread application is the IDEO-inspired Design Thinking Method [44], as it is spreading from Stanford University's d.school since the mid-2000s [45]. Founding on human-centered design principles, aka needs, Design Thinking tools guide the designer (business and technical developer) in iterative cycles through five overlapping phases [45]:

- *Empathize*: Empathy is the foundation of human-centered design. The problems you are trying to solve are rarely your own, they are those of particular users. Build empathy for your users by learning their values.
- *Define*: The define mode is when you unpack your empathy findings into needs and insights and scope a meaningful challenge. Based on your understanding of users and their environments, come up with an actionable problem statement: your Point Of View.
- *Ideate*: Ideate is the mode in which you generate radical design alternatives. Ideation is a process of "going wide" in terms of concepts and outcomes a mode of "flaring" instead of "focus." The goal of ideation is to explore a wide solution space both a large quantity and a broad diversity of ideas. From this vast repository of ideas, you can build prototypes to test with users.
- *Prototype*: Prototyping gets ideas out of your head and into the world. A prototype can be anything that takes a physical form a wall of post-its, a role-playing activity, and an object. In early stages, keep prototypes inexpensive and low resolution to learn quickly and explore possibilities.
- *Test*: Testing is your chance to gather feedback, refine solutions, and continue to learn about your users. The test mode is an iterative mode in which you place low-resolution prototypes in the appropriate context of your user's life. Prototype as if you know you are right but test as if you know you are wrong.

Design thinking methods help ensuring that a new product but also services and software [46] is truly needed and wanted, and its specifications and processes fit the actual user demands. It thus increases the potential for market success.

Many startups adopt the lean startup idea in their own ways. For software development, it is common to realize the Build–Measure–Learn loops as development iterations. In QuickNews, lean startup methodology is realized by a continuous process of testing business hypotheses based on Scrum. For each iteration, the CEO, CTO, and relevant software developers did a Sprint planning meeting that defined a business hypothesis. MVP was created in the Sprint to validate the hypothesis. While it is straightforward to perform the Sprint, it takes time for the CEO and CTO to consolidate lessons learned from each Sprint. We illustrated for three Build–Measure–Learn loops as the result of a postmortem analysis as shown in Table 3.

9 Conclusions

Since the release of Shane and Venkataraman's famous article "*The promise of entrepreneurship as a science*" in 2001, a host of new and modern publications have come out describing an entrepreneurial persona or archetype. This persona is behaving radically differently from the rational financial actor in classic economics.

Firstly, the entrepreneur is most often starting with a resource view rather than a market view [33]. This means that the starting point of entrepreneurship is to "*find and exploit opportunities*" based on entrepreneurs' own skills, assets, and network rather than looking for opportunities based on imperfections in the market. Secondly, the entrepreneurs' main tools seem to be the creativity of effectuation rather than causal analysis. The entrepreneur starts with only a broad set of overall personal values rather than a clear rational objective, and more often a starting point for improvisation than a clear plan. Instead of access to unlimited resources, the entrepreneur has to "make do" with what he has and find ways to reassign resources to new and creative applications. Finally, the entrepreneur manages with a loose and constantly changing business model rather than an iron-clad strategy and business plan—and that business model goes through a series of build–measure–learn iterations to constantly improve on the entrepreneurial project. This paints a picture of a creative improviser rather than a rules-bound rational actor. We seem to be playing Jazz instead of Symphonies.

And these articles have often not been satisfied with quietly describing how an entrepreneur *does* behave. They are often suggesting how entrepreneurs *should* behave. The suggestion is that in many instances building business based on the company's core competence is a good idea. The same goes for basing the project on a RBV. The analysis is that acting according to the models of Bricolage and Effectuation in many instances will increase the success rate of entrepreneurs. Business modelling and Lean Startup certainly suggest their models as normative structures.

This consequently might be a part of bridging the gap between academics and entrepreneurs. If we viewed these different normative suggestions together, there may be a basis for getting closer to answering Shanes' quest for a subset of entrepreneurial subprocesses.

We of course do not claim to cover the total range of entrepreneurial theory written in our millennium. Over the past two decades, a number of different theoretical perspectives have emerged to describe the logic and behavior underlying the entrepreneurial process. While it would be interesting to link all recent entrepreneurial theories, this work focuses on six fundamental pillars of theory that are often used by entrepreneurs and their advisors, namely RBV and core competence; effectuation and causation; bricolage; entrepreneurial opportunities; business model innovation; and lean startup. They are all grounded in the Opportunity-Based Entrepreneurship theory, and the Resource-Based Entrepreneurship theory.

As mentioned by Kurt Lewin, "there is nothing more practical than a good theory" [47], we believe that entrepreneurs benefit from the awareness of these entrepreneurial theories and understanding how they can be connected not only to business decisions but also to engineering activities. Entrepreneurs can apply these principles and recommendations to appropriate circumstances during their entrepreneurial journey. For researchers in the entrepreneurship area, this is also among the attempts bringing the thick body of knowledge in the field to practical and technological domains.

References

1. Forenklingsudvalget for Erhvervsfremme: Fokuseret og fremtidssikret (2018)
2. Landström, H.: The roots of entrepreneurship research. The intellectual development of a research field. N. Engl. J. Entrep. **1**(2), 9–20 (1999)
3. Cantillon, R.: Essay on the Nature of Trade in General. Henry Higgs, London (1755). (edition and translation 1959)
4. Schumpeter, J.A.: Capitalism, Socialism, and Democracy. Harper & Row, New York (1942)
5. Schumpeter, J.A.: The Theory of Economic Development. Harvard Economic Studies, Cambridge, MA (1934)
6. Carlsson, B.: The rise of small business: causes and consequences. In: Adams, W.J. (ed.) Singular Europe: Economy and Polity of the European Community After 1992, pp. 145–169. University of Michigan Press, Ann Arbor, MI (1992)
7. Carlsson, B., Acs, Z.J., Audretsch, D.B., Braunerhjelm, P.: Knowledge creation, entrepreneurship, and economic growth: a historical review. Ind. Corp. Chang. **18**(6), 1193–1229 (2009)
8. Gartner, W.B., Shane, S.A.: Measuring entrepreneurship over time. J. Bus. Ventur. **10**, 283–301 (1995)
9. Shane, S.: Reflections on the 2010 "AMR" decade award: delivering on the promise of entrepreneurship as a field of research. Acad. Manag. Rev. **37**, 10–20 (2012)
10. Shane, S., Venkataraman, S.: The promise of entrepreneurship as a field of research. Acad. Manag. Rev. **25**(1), 217–226 (2000)
11. Amit, R., Glosten, L., Muller, E.: Challenges to theory development in entrepreneurship research. J. Manag. Stud. **30**(5), 815–834 (1993)
12. Simpeh, K.N.: Entrepreneurship theories and empirical research: a summary review of the literature. Eur. J. Bus. Manag. **3**(6), 1–8 (2011)
13. Barney, J.: Firm resources and sustained competitive advantage. J. Manag. **17**, 99–120 (1991)
14. Wernerfelt, B.: A resource-based view of the firm. Strateg. Mark. J. **5**, 171–180 (1984)
15. Prahalad, C., Hamel, G.: The core competence of the corporation. Harv. Bus. Rev. **68**, 79–91 (1991)
16. Sarasvathy, S.: Causation and effectuation: toward a theoretical shift from economic inevitability to entrepreneurial contingency. Acad. Manag. Rev. **26**, 243–263 (2001)
17. Sarasvathy, S.D., Venkataraman, S.: Strategy and entrepreneurship: outlines of an untold story. In: Hitt, M.A., Freeman, E., Harrison, J.S. (eds.) Strategic Management Handbook, pp. 650–668. Blackwell, Oxford (2000)
18. Baker, T., Nelson, R.E.: Creating something from nothing: resource construction through entrepreneurial bricolage. Adm. Sci. Q. **50**, 329–366 (2005)
19. Maurya, A.: Running Lean. O'Reilly, Sebastopol, CA (2012)
20. Osterwalder, A., Pigneur, Y., Tucci, C.L.: Clarifying business models: origins, present, and future of the concept. Commun. AIS. **15**, 751–755 (2005)
21. Osterwalder, A., Pigneur, Y.: Business Model Generation. Wiley, Hoboken, NJ (2010)
22. Zott, C., Amit, R., Massa, L.: The business model: recent developments and future research. J. Manag. **37**, 1019–1042 (2011)
23. Blank, S.: The Four Steps to the Epiphany—Successful Strategies for Products That Win. K&S Ranch, Menlo Park, CA (2005)
24. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses, 1st edn. Currency, New York (2011)
25. Gartner, W.B.: Who is the entrepreneur? Is the wrong question. Am. J. Small Bus. **12**, 11–32 (1988)
26. Pinchot, G.: Who is the intrapreneur? In: Intrapreneuring: Why You Don't Have to Leave the Corporation to Become an Entrepreneur, pp. 28–48. Harper & Row, New York (1984)
27. Hammersley, M., Atkinson, P.: What is ethnography. In: Ethnography: Principles in Practice, pp 9–25. Routledge, London (1995)

28. Jorgensen, D.L.: Participant observation. In: Scott, R.A., Kosslyn, S.M. (eds.) Emerging Trends in the Social and Behavioral Sciences. Wiley, New York (2015)
29. Porter, M.E.: Competitive Strategy. Free Press, New York (1980)
30. Teece, D., Pisano, G., Shuen, A.: Dynamic capabilities and strategic management. Strateg. Manag. J. **18**(7), 509–533 (1997)
31. Eckhardt, J.T., Shane, S.A.: Opportunities and entrepreneurship. J. Manag. **29**(3), 333–349 (2003)
32. Gangi, J.: The synergies of artistic and entrepreneurial action. J. Arts Manag. Law Soc. **45**, 247–254 (2015)
33. Dahle, Y., Anh, N.D., Steinert, M., Chizhevskiy, R.: Resource and competence (internal) view vs. environment and market (external) view when defining a business. In: 2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), pp. 1–9 (2018)
34. Schumpeter, J.A.: Theorie der wirtschaftlichen Entwicklung. Duncker und Humblot, Berlin (1912)
35. Simon, H.A.: The Sciences of the Artificial, 3rd edn. MIT, Cambridge, MA (1996)
36. Nguven-Duc, A., Dahle, Y., Steinert, M., Abrahamsson, P.: Towards understanding startup product development as effectual entrepreneurial behaviors. In: Felderer, M., Fernández, D.M., Turhan, B., Kalinowski, M., Sarro, F., Winkler, D. (eds.) Product-Focused Software Process Improvement, pp. 265–279. Springer International, Cham (2017)
37. Levi-Strauss, C.: Ther Savage Mind. University of Chicago Press, Chicago, IL (1967)
38. Ritter, T., Lettl, C.: The wider implications of business-model research. Long Range Plan. **51**, 1–8 (2018)
39. Baden-Fuller, C., Morgan, M.S.: Business models as models. Long Range Plan. **43**, 156–171 (2010)
40. Rambow-Hoeschele, K., Nagl, A., Harrison, D.K., Wood, B.M., Bozem, K., Braun, K., Hoch, P.: Creation of a digital business model builder—a concept to simulate a digital twin of a business model and its imperative nature. In: 2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC) (2018)
41. Morris, M., Schindehutte, M., Allen, J.: The entrepreneur's business model: toward a unified perspective. J. Bus. Res. **58**, 726–735 (2005)
42. Moore, G.: Crossing the Chasm. HarperCollins, New York (1995)
43. Ghezzi, A.: Digital startups and the adoption and implementation of lean startup approaches: effectuation, bricolage and opportunity creation in practice. Technol. Forecast. Soc. Chang. **146**(C), 945–960 (2019)
44. Brown, T., Katz, B.: Change by design. J. Prod. Innov. Manag. **28**(3), 381–383 (2011)
45. Design Thinking Bootleg: Stanford d.school website (n.d.). https://dschool.stanford.edu/resources/design-thinking-bootleg. Accessed July 2019
46. Jensen, M.B., Lozano, F., Steinert, M.: The origins of design thinking and the relevance in software innovations. In: International Conference on Product-Focused Software Process Improvement, pp. 675–678. Springer, Cham (2016)
47. Lewin, K.: Psychology and the process of group living. J. Soc. Psychol. **17**, 113–131 (1943)

Pivoting in Software Startups

Sohaib Shahid Bajwa

Abstract To be able to handle intense time pressure and survive in dynamic markets, software startups make critical decision constantly on whether to change directions or stay on chosen path. This is known as to pivot or to persevere in terms of lean startup. Pivot is an essential and crucial activity for software startups to initially survive, and then grow and eventually obtain a sustainable business model. We find several examples of software startups (Twitter, Instagram, etc.) that successfully pivoted during their entrepreneurial journey and became successful software companies. There is a lack of empirical evidences on why and how software startups make pivots. This chapter provides reader a better understanding on pivots, different types of pivots, and factors that trigger pivots. We employ case study and case survey methods. The results show that customer need, customer segment, product zoom-in, and technology pivots are the major types of pivots software startups have made. We also identify several new pivot types including market zoom-in, market zoom-out, complete, and side project pivots. Negative customer reaction and flawed business model are the prominent factors triggering pivots. The findings provide practical knowledge to software startups and practitioners, which they can utilize to better understand and perform pivots.

Keywords Software startup · Pivot · Empirical study

1 Introduction

Most of us are aware that Twitter is one of the famous examples of a successful startup. It is a very popular microblogging platform. Only few people are aware that it was originally started as a podcast service provider back in 2005 [1]. Another example of a startup is Instagram. Initially, it was a social check-in application called

S. S. Bajwa (✉)
University of Calgary, Calgary, AB, Canada
e-mail: sohaibshahid.bajwa@ucalgary.ca

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_2

Burbn, combining feature of a Foursquare (a photo share app) and a game called Mafiawars [2]. Now it is a popular photo- and video-sharing social networking service. These examples show that even famous startups struggle to get their product or business right immediate, and do not end up with what they initially started.

Startups need to produce cutting-edge products and grow rapidly under the condition of extreme uncertainty. It imposes different challenges that are very diverse in nature and can be related to product, finance, market, or about the entrepreneurial team itself. Developing cutting-edge product, defining a minimum viable product (MVP), acquiring first paying customers, and building entrepreneurial team are some of the significant challenges that software startups face [3].

In order to address these challenges, and eventually obtain a sustainable business model, startups change their direction relentlessly. This change in direction is termed as "*Pivot*" in Lean Startup approach [4]. According to Ries [4], "*A pivot is a strategic change, designed to test a fundamental hypothesis about a product, business model or engine of growth*."

It is observed that startups avoid pivoting when needed which leads to failure as described in [5]. It is also a most frequently occurring commonality among different successful startups [4]. Pivot is essential for a startup to survive, grow, and eventually obtain a sustainable business model. Flickr also provides one such example where startup pivoted and became a successful company. Flickr used to be an online multiplayer role-playing game rather than a photo managing and sharing service [2].

Pivot is arguable one of the key elements of any software startup. However, there is little rigor and relevance existing in the literature regarding software startups and especially about pivots, due to the emerging nature of startup research [6, 7]. There is a scarcity of empirically evidenced knowledge on pivoting in software startups, especially what trigger software startups to pivot, how and why they make certain pivot decisions, and how they actually pivot.

In order to address the knowledge gap in research, we formulate the following research questions:

- RQ1. What are the factors that trigger software startups to pivot?
- RQ2. What are the different types of pivots software startups have made during their entrepreneurial journey?

This chapter primarily addresses this knowledge gap by providing researchers and practitioners with empirical evidences on different factors triggering pivots and the types of pivot startups face, increasing the chance of achieving startups objectives.

Main findings

- ✓ **Startups pivot more than once:** In order to survive, grow, and eventually obtain a sustainable business model, startups pivot more than once.
- ✓ **Most occurring type of pivot:** Pivots can be of many types, however, customer need pivot is the most common type of pivot.
- ✓ **Other types of pivots:** Pivot can be related to product too such as product zoom-in and zoom-out.
- ✓ **New pivot types: Complete, side project, and team:** The empirical evidence also proposes new pivot types—complete pivot, side project, and team pivot.
- ✓ **Negative customer reaction:** Negative customer reaction is the major triggering factor causing software startups to pivot.
- ✓ **More than one factors triggering pivot:** It is also possible that there are more than one triggering factors involved in pivoting.

2 What Researchers Say About Pivot?

This section presents different definitions presented in literature about pivot and also describes the characteristics of pivots. Different models of software startups' evolution are also discussed. It also presents the work related to pivot types.

2.1 Definition and Characteristics of Pivots

It is not clear from the literature who initially invented the term, Pivot. Ries described pivot in his book [4] and from there it gained popularity. Ries used the analogy from the basketball sports, where pivoting player keeps one foot planted, while moving the other. Startups also exhibit a similar behavior.

Many people generally believe that pivot is synonym of change; however, this understanding is not correct. Pivot is not just about introducing any change and making any decision. Ries defines pivot as a special kind of change designed to test and validate the assumptions about a product, business model, and the engine of growth [4]. Bajwa [8] defines pivot as:

> A strategic decision which leads to the significant change to one or more, but not all, elements of a startup: product, entrepreneurial team, business model or engine of growth.

It is to note that when all of these elements change at the same time, it is not considered a pivot but starting a completely new and different business.

Software startups pivot to avoid wasting critical resources. Pivot allows a startup to spend its energy and limited resources to pursue the direction that is most promising to create business value and eventually leads to a sustainable business model.

2.2 Possible Models of Startup Evolution and Pivots

A software development model for early-stage software startups helps software startups in their decision-making, e.g., when to abandon an idea or when to move forward with this [9]. However, the model does not provide much information about pivots. There is another evolutionary model for early-stage software startups proposed that highlights the significant pivots that software startups have taken during their entrepreneurial journey [10].

Pivoting decision and architecture decisions can be combined too as described in [11]. The study presents the similarities and differences between these two types of decisions: pivoting decisions and architecture decisions. Risk is one of the common triggering factors in making a decision both about pivots and about architecture. In a story of a software startup which pivoted, the story described the failures of the startup, the lesson learned, and how the startup pivoted to develop a new product which eventually got the attention of the medium and large enterprises [12].

A co-relation exists between types of pivots and different phases of startup's life cycle. The authors present that pivot can occur at any stage during a startup's life cycle. Therefore, it is important to study pivot at different product development stages [13].

2.3 Software Startups and Types of Pivots

There are couple of studies conducted to understand the types of pivots. In one such study, authors present the relationship between different pivot types (product zoom-out, customer segment, business architecture, etc.), and how they affect the different parts of the lean canvas model [14]. In another study, authors provide an overview of how leading software companies, e.g., Twitter, Google, and Facebook pivoted historically. The conclusion is that most successful startups have made multiple pivots during their journey [15].

2.4 Role of Minimum Viable Product and User Experience in Startups

There are several other studies conducted to explore different concepts that are associated with pivots. For example, Münch et al. [16] describe the industry–academia collaboration to create MVP. Moreover, there are several studies recently conducted to explore user experience in software startups [17, 18].

3 Our Methodology

We employ case survey and case study methods to explore pivoting in software startups.

Case Survey Firstly, we conducted case survey method to identify different triggering factors causing pivots and the types of pivots. We used secondary data available online for the purpose of data collection of our case survey study. There are several sources of secondary data such as magazines, census, blogs, and reports. The advantages of using secondary data are that data collection process can be inexpensive and fast [19]. When used with care and diligence, secondary data can provide an efficient way of getting initial understanding about the questions under investigation. It is also often helpful in designing subsequent primary research and can provide a baseline with which to compare the primary data analysis results [20]. We adopted Systematic Literature Review (SLR) guidelines presented by Kitchenham [21] to ensure our data collection and analysis process is more systematic and reliable. The implementation details, how we conducted the data collection, and analysis process can be found in [8].

Case Study Case study approach is defined as "an empirical inquiry that investigates the contemporary phenomena within real-life context" [22]. The case study was performed to identify the factors triggering pivots and the types of pivots. We selected seven software startups, pivoted during their entrepreneurial journey. We conducted two rounds of semi-structured interviews for the purpose of data collection. All the interviews lasted from 30 to 60 min. The interviews were transcribed verbatim for analysis purposes. Each interview involved at least one of the founders who were part of pivot decision-making process and knew the entrepreneurial journey of the startups. We performed both within case analysis and cross-case comparisons for data analysis purpose.

4 Results

The chapter describes in detail the results about work related to pivoting in software startups. In the first section (Sect. 5.1), case survey results are discussed, while the second section (Sect. 5.2) describes the results of the case study.

4.1 Case Survey Results

We included 49 software startups in our case survey study. These startups came from all over the world; however, majority (37 startups) is based in the USA while four are from Canada. There were two startups from Israel, while the other four are located in the UK, Australia, New Zealand, and India. We could not get the information about two companies' geographical location.

Following are the major business domains:

- Social networks (30.61%)
- E-commerce (24.44%)
- Finance and business (12.24%)

More contextual information about 49 software startups included in our study can be obtained in [8].

We analyzed the data obtained from 49 software startups. We were mainly looking to identify the factors triggering pivot in software startups and the pivot types. Table 1 describes the triggering factors, internal or external, its description, and # of pivots caused due to these particular factors.

The major pivoting types identified in the 49 cases are listed in Table 2, organized under the dimensions of "product," "market," and "others". (Note that our findings did not reveal any pivot that can be classified as financial or team related pivots.) One pivot instance is classified under one pivot type only.

Next, we discuss different pivot types with an illustrative example of a startup.

Product—Zoom-In Pivot Flickr provides an example of a zoom-in pivot. It originally started as an online massive multiplayer role-playing game called Game Neverending. However, it failed to attract the customers' attention. The game had a feature that provided a photo-sharing tool to allow players to share photos and save them on a webpage while playing. This turned out to be the most popular feature of the game. The founders decided to leverage this popularity and pivoted toward a photo-sharing application now known as Flickr.

Product—Technology Pivot Wix initially started as a Flash-based website builder before 2011 when Flash was the best option available for website development. With the emergence of smartphones, mobile devices, and HTML5, Flash was not anymore a viable option for their business because of poor performance with the smartphones. Due to this reason, Wix pivoted toward providing the website

Table 1 Major factors triggering pivots in software startups [8]

Triggering factors	Internal or external	Description	Pivot instances (#)
Negative customer reaction	External	It refers to slow customer acquisition, slow customer retention, no or negative response from customers, etc.	15
Unable to compete with competitor	External	Several competitors (e.g., big companies, other startup companies) outplay the startup by working on the same idea more effectively.	5
Technology challenge	External	Several challenges related to technology including limitation with existing technologies (e.g., performance issues), better technology availability due to emergence of disruptive technologies.	5
Influence of investor/mentor/partner	External	Suggestion or pressure from investors, mentors, or partners to change the direction.	4
User appreciation of one particular feature of the product	External	Users appreciate one specific feature, rather than showing interest in the whole product.	4
Unanticipated use of product by users	External	Users use product in an unexpected manner, which was not foreseen before.	4
Wrong timing	External	Providing a solution for which market is not yet ready to accept.	3
Positive response from an unforeseen customer segment	External	Among different customer segments, one specific segment shows more interest in the product.	3
Running into legal issue	External	Legal problems with other companies (e.g., copyright issues).	1
Side project more successful than main project	External	Lack of interest from customers in the main project, but they are interested in the side project.	1
Targeted market narrowing	External	The initially targeted market becomes smaller for the startup to survive and grow.	1
Flawed business model	Internal	High cost of customer acquisition or revenue model is not working.	7
Identification of a bigger customer needs through solving an internal problem	Internal	While developing a solution internally, to support the core product, the startup realizes that the identified internal problem is the real pain point for the customers, compared to the problem their original product solves.	5
Unscalable business	Internal	Solving a problem in which not many people are interested, resulting in unscalable business.	5

Table 2 Major pivot types in the software startups

Dimension	Pivot type	Pivot instances (#)
Product	Zoom-in: a single feature becomes the whole product.	7
	Technology: the same solution using different technology.	5
	Platform: a product becomes a platform or vice versa.	3
	Zoom-out: a whole product becomes one feature of a larger product.	2
Market	Customer need: switching to a different problem that customers have.	17
	Customer segment: switching to a different customer segment than the one originally conceived.	6
	Channel: finding a more effective way to reach the customers	1
	[a]Zoom-in: Focusing on one specific market sector rather than the whole market.	1
Others	[a]Complete: Significant change in product, market, and financial dimensions but the entrepreneurial team remains the same.	11
	[a]Side Project: A different business idea parallel and unrelated to the main project becomes the main project.	2
Total		55

[a]New pivot types

development platform using HTML5. Technology challenge was the triggering factor.

Product—Platform Pivot Platform pivot can be bidirectional by definition, either from a particular product to an underlying platform or vice versa. appMobi pivoted from product to platform. appMobi, originally called Flycast, pivoted from a mobile app for iPhone, Blackberry, and Android to a set of tools that support cross-platform development of mobile apps.

Market—Customer Need Pivot Yelp and Hopper are the two startups that discovered real customer needs due to the unanticipated use of their products by users. For example, Yelp intended their emailing system to be used by users to connect with others for recommendations about local businesses. However, the users started to use the system to write reviews about the local business. This emergent new mode of using the system soon caught the attention of the founders, through which they identified a different yet more promising need to be addressed, and pivoted toward providing reviews about local businesses.

Market—Customer Segment Pivot Groupize is a representative case of market—customer segment pivot. It originally targeted at consumer business by generating demands through creating white-label agreements with multiple travel agencies. They were unable to compete with their competitors in the travel management service areas, and hence shifted their focus toward providing a group meeting solution for hotels, meeting planners, and travel management companies.

Market—Channel Pivot Site59 presents an example of channel pivot. Their initial idea was to create mini-vacation packages by combining last-minute deals from different air travel, hotel accommodation, and other travel-related services and websites. However, the idea did not go viral as they expected, and the customer acquisition rate was very low. One of the investors suggested Site59 to change the distribution channel to reach customers, using the business-to-business-to-consumers (B2B2C) model. Following the suggestion of their investor, Site59 pivoted the channel to reach their customers and their new service was to prepare last-minute vacation packages for different airlines and vacation portals.

Market—Zoom-In Pivot >Ignighter is an interesting case of such a pivot. The primary aim of Ignighter was to develop a dating website for different US users mainly. The founders expected to receive positive response from the USA, their home market. Unexpectedly, the idea got promising attraction from customers in Asian markets, especially in India. The founders carefully analyzed the demographic data, and identified the promising user growth in India as compared to any other country. As a result, the founders decided to focus on this market segment only and made the pivot to officially become Indian dating site.

Complete Pivot A complete pivot is a pivot when an entrepreneurial team has to come up with a new and innovative idea after their initial product/idea was failed or outplayed due to different factors, e.g., big companies started working on that idea and attracted their niche markets as they had more resources. This pivot implies significant changes in many aspects of a startup, including product, targeted market, and finance. The only unchanging element is the entrepreneurial team that carries on the learning from the past experience to set the new directions.

Twitter is a representative case of complete pivot. It was initially started as Odeo, a podcast service to allow sharing and recording of different podcasts. Then Apple iTunes started to fill this gap as they had more funds and resources, leaving behind the Odeo service. Odeo founders were unable to compete with Apple iTunes; the startup team did brainstorm to do decide what to do next and found a new direction called Twitter.

Side Project We define Side Project as it is a special kind of project that runs parallel to the main project of a software startup, but may be based on a different even unrelated business idea and target at a different set of customers. Groupon is a famous example of side project pivot where side project outshines the main one. Groupon initially started as "The Point": social campaigns to collect funds for good causes. Campaigns were only successful when a certain tipping point was reached. However, this project did not get much user traction, and there was no clear revenue model. Due to these factors, it became very difficult to generate sufficient revenue from this idea and to achieve breakeven. However, the side project the team started in parallel, using the same tipping point but for group buying and local deals, attracted more users. Eventually, the side project took off and it has now become the daily deal website famously known as Groupon.

Multiple Pivots It is possible that startups do multiple pivots during their entrepreneurial journey. Instagram and Retention Science provide examples of multiple pivots. Take an example of Retention Science. It initially started as providing independent artists a platform where they could promote niche brands and products via social media. Although the founders contacted different channels such as working with YouTube celebrities, sponsoring local concerts, but their business proved to be unscalable. They also discovered that the customers were reluctant to appear as sellout by promoting different brands. The unscalable business and negative customer reaction triggered their first complete pivot. They pivoted toward providing a social media-based analytics and referral platform for e-commerce businesses. The second complete pivot happened due to the competitors. The founder discovered that there were many well-funded startups already working in the same area, and there were little chance that they could acquire the funds to compete and become successful. Without funding, they could not accelerate their product development and increase user growth, and hence unable to compete with their competitors. Due to these triggering factors, they pivoted completely again toward a retention automation platform that used artificial intelligence techniques, to engage customers and increase customer retention.

4.2 Case Study Result

We selected seven software-based startups that were at different product development stages from concept to mature product. All of these seven startups pivoted during their product development. The contextual information and case description can be found in [23].

The summary of the pivots that occurred in these seven software startups including pivot types, factors causing pivots, and the corresponding product development stages is presented in Table 3.

Startups make pivots while they are at the conceptual stages or have very limited customers. Team pivot and market—zoom-out are two new types of pivots identified. It is also possible that multiple pivots occur instantaneously.

Team Pivot Team pivot was also manifested in the case of Hooka. While developing the working prototype in 2011, during an investor meeting where they were supposed to presenting their prototype, they failed to give the demo of their prototype because it did not work. They changed their whole development team, and hired new professionals who could develop.

Table 3 Summary of pivots in software startups

Startup	Before pivot	After pivot	Pivot type	Triggering factor	Product development stage
Dicy	Online community platform for entrepreneurs	Providing video service facilities to startups	Customer need	Negative customer reaction	Functional product with limited users
TicketGo	Use of NFC to validate event tickets	Use of QR code to validate event tickets	Technology	Targeted market narrowing	Concept
OneWeb	A web aggregator for users	A web aggregator for companies	Customer segment	Targeted market narrowing	Functional product with limited users
DocMine	An encrypted software	A unified API to access social media sources	Complete	Unable to compete with competitor	Functional product with limited users
	Focusing on private markets and companies	Focusing on developers only	Customer segment	Negative customer reaction	Functional product with limited users
	One of the founders left	The founder joined back	Team	Team competencies	Working prototype
Hooka	Sell magnetic tapes	Bidding sys. for nightclubs seats	Complete	Negative customer reaction	Functional product with limited users
	Bidding sys. for nightclubs seats	Ticket validation sys. for event org.	Customer segment	Negative customer reaction	Concept
	SMS-based application	A complete ticket validation sys.	Zoom-out product	Negative customer reaction	Concept
	Low competent developers	Hired new developers	Team	Negative customer reaction	Working prototype
Club-net	Educational videos	Interactive app	Technology	Negative customer reaction	Functional product with high growth
	Interactive app	Social network community	Zoom-out product	Negative customer reaction	Functional product with high growth
EasyLearning	Game-based learning platform to be used on Sony phones	Game-based learning platform to be used on iPhone	Technology	Technology challenge	Working prototype
	Game-based learning platform to be used on iPhone	Use of java and web servers	Technology	Negative customer reaction	Working prototype
	A software to focus on university students	Elementary schools, org. with training demands	Zoom-out market	Targeted market narrowing	Functional product with high growth

5 Conclusion

This section discusses the pivot types and factors triggering pivot. It also includes the implication of the results to practitioners.

5.1 Major Pivot Types

Retention ScienceCustomer need pivot (17 out of the 55 pivot instances) is the most common pivot type identified in case survey. This is not surprising in the sense that it is consistent with the nature of a software startup. While working with highly dynamic environment, rapidly changing technologies and pressure to develop innovative products, software startups strive hard to identify and solve the real and unique customer problems that are worth solving. In order to better understand the real customers' problems, software startups need to pivot relentlessly. From the case survey study, we see that Yelp pivoted subsequently according to different customer needs they discovered.

In the course of better understanding the market, software startups often discover that even though the problem they want to solve is real, it is not the problem of the customer segment they have initially thought. Six pivot instances made the customer segment pivot, the second most common market-related pivot type that our case survey study has identified. It is possible that startups are solving a real problem; however, their initially perceived customers may not be interested in that problem. The challenge for startups is that they may not know their future customers in advance, and risk spending too much time and resources to come up with a product that fails to achieve product/market fit. However, this learning from failure can be helpful to identify new targeted customers and then pivot toward them as mentioned in DocMine and OneWeb.

Pivots specifically related to product are also important. Seventeen pivot instances in our case survey sample are related to product. Software startups have to reconsider their products and different features in order to find the problem/solution fit and/or product/market fit. This often leads toward product-related pivots. Among different product-related pivots identified by study, the zoom-in pivot is relatively more common than other product-related pivot types. It often happens that customers are more interested in one or two particular features rather than the whole product. Pinterest and Flickr are good examples of product zoom-in pivot. Ideally, instead of wasting resources and building a complex product with lots of features, it is better to focus on one feature that actually gained the attractions of the customers and build it first. However, it is not easy to understand which can be the valuable feature to build first. MVP can be helpful in this regard. By building MVPs, entrepreneurs have an initial set of features that are appreciated by their initial users.

The opposite of product zoom-in pivot is product zoom-out pivot, which reflects the need for achieving the problem/solution fit. It is possible that software startups have identified the right set of problems, but their products are still incomplete. They need to expand their solutions to add more features as manifested in the case of Hooka.

Another important product-related pivot type is technology pivot, second most common within the product dimension, which reflects the role technology plays in software startups. Dynamic and rapidly changing technologies are always challenging for software startups as they are prone to technology pivots due to the fact that they are building technology-intensive products. Often technology pivots are driven by the need of software startups to be always at the cutting-edge of technological advancement. This is manifested in the cases of ClubNet, EasyLearning, TicketGo, Wix, and Instagram.

In order to support their products and make solutions complete, software startups often develop both products and supporting platforms. In addition to supporting the core product, sometimes it happens that the platforms solve larger problems than their original products do. Therefore, platform pivots are desired. appMobi is one such example. It is also worth mentioning that platform pivots can depart from a hosting platform to specific products running on the platform. However, we could not find evidence in our case collection. It is possible that this direction is not as frequent as the product to platform direction.

In terms of the scope of change and the amount of effort and resource, complete is the most demanding pivot. This is a new pivot type identified in our study. It is the second most common among all pivot types (11 out of 55 pivot instances). We term it complete pivot since it is related to almost all the aspects of a startup, including product, market, and financial, with only the original team as the rooting element in the pivot, which ensures that the learning from previous failing experience is maintained and startups learn from those failed experiences. Famous companies such as Twitter went through significant changes in their business during their entrepreneur journey before they found successful and sustainable business model to scale.

Among the newly identified pivot types, side project pivot is another interesting new pivot type. Even though working under a highly pressurized and extreme chaotic environment, many software startups may decide to run one or more side projects simultaneously. These side projects are generally not related to their core ideas. We define side project as it is a project that runs parallel to the main project, but may target at a different set of customers. These side projects may become main projects when outshining them. Groupon is a good example, initially started as a side project.

The third new type is market zoom-in pivot, which is demonstrated by the Ignighter case. It is a reflection of striking the product/market fit. It is often suggested that, to start with, a startup should find its focus and niche market, identify the early adopters of their product. This type of pivot shows the need to do so.

Another finding worth mentioning is multiple pivots which can happen either simultaneously or separately. Some pivots may be closely linked and there is a possibility that chain reaction occurs, which means one pivot triggers several other pivots, known as "the domino effect" [14]. Instagram is one such example. One startup may have several pivots spread across their courses of development too, which is revealed in the case of Retention Science (two separate complete pivots). This indicates the importance of constantly checking and correcting the directions until a startup obtains a sustainable business model.

In [3], authors reveal that building an entrepreneurial team is one of the prominent challenges for software startups. As a response to this challenge, the entrepreneurial teams go through significant changes in team composition. This kind of pivot is termed as a team pivot. The change can be related to the inclusion of a new key member (e.g., co-founder) or having a new development team completely. Both Hooka and DocMine evidenced team pivot during their journeys.

5.2 Factors Triggering Pivots

The majority of the triggering factors are external factors. It means events that are occurring beyond the control of a startup. This implies that for many of the studied startups, major pivots they made were primarily a reaction to what happened externally rather than purposefully design change as suggested by Ries' definition of pivot [4].

Negative customer reaction is the most common factor triggering software startups to pivot. Slow user acquisition, limited user retention rate, and no growth are some manifestations of negative customer reaction. In the case of Dicy, the startup reacted to negative customer reactions and pivoted accordingly.

In order to come up with innovative and cutting-edge products, software startups have to compete with other competitors, especially with big companies. These big companies have much more resources in terms of skilled people and funds than what software startups can wield. Due to these resources, they can develop innovative products rather quickly as compared to startups. Twitter stumbled upon this challenge when their initial idea of offering podcast services was taken by Apple who then outplayed Twitter with the launch of iTunes. Twitter pivoted drastically.

Software startups is known for being heavily influenced by stakeholders and investors [7]. The suggestions from the investors/mentors/partners significantly affect the development processes of software startups, and they may eventually change their course. It is probable that a software startup has a good technological and innovative idea, but their investors, mentors, or partners have a different vision, which affects the overall direction of the startup. Site59 is a good example of this.

Meanwhile, our case survey findings also reveal that a flawed business model is a significant internal factor that triggers software startups to pivot. Scaling a business that has flaws in their business model, or in other words, "pre-matured scaling" [5], may lead to the eventual failure of a startup. It is very crucial to first correct your business model especially before doing scaling. Software startups may avoid failure by identifying the flaws in their business model earlier and pivot accordingly. There are some indications of flawed business models such as low or no revenue, or high acquisition cost. It was demonstrated in the cases of Groupon.

There is no linear and one-to-one mapping between pivot types and factors triggering pivots. It is difficult to conclude that a certain pivot type is always due to a specific triggering factor. Unanticipated use of product by users is a factor that triggers Pinterest to product zoom-in pivot, while in the case of Yelp, the same factor causes it to pivot to different customer need.

Lack of competencies (identified by case study) is one factor causing software startup teams to pivot, as exhibited in the case of Hooka. They fired their whole development team because of a lack of technical skills, hired new professionals, and developed the product from scratch. It is also possible that one of the co-founders initially left and later on joined. DocMine evidenced the team related pivot, when their co-founder, who earlier left, came back and rejoined the team.

There is a debate that how software startups are different than any other startup. Our findings clarify this debate and allow us to reflect upon the role of the unique nature of software product plays in software startup pivoting. For example, due to the flexible and modifiable nature of a software product, product zoom-in and zoom-out pivots should be relatively easy to implement for software startups than for startups that produce physical products, such as hardware or medical devices.

5.3 Implications

Our findings have direct implication for software startup practitioners. The empirical evidences from the findings suggest that software startup teams should collect maximum knowledge while living in chaotic and uncertain situations. Considering the chaotic, dynamic, and unpredictable environment of software startups, the validated learning will be crucial to driving business and product decisions in order to proceed in the right direction. They should use their previous experiences, learning from failures and then set the direction to become a successful software company.

Our findings also help startup practitioners to make informed decision. Both the identified pivot types and triggering factors can be utilized by startups to make more informed decisions on when and how to pivot. It also implies the use of different metrics to track customer reaction, product usage, etc., to support informed decisions regarding pivoting. It is very crucial to get the actionable metrics, especially in the case of product zoom-in pivot, where data analytics can help to identify the usage of different features. We should not spend time and energy in collecting metrics that we cannot use or that has no value. One can identify the most frequent features used by users and then make product simpler and at the same time, focusing on the most usable and valued feature.

Software startups generally work on one product idea at a time. However, our findings show the importance of side project. Our study provides empirical evidences that there are software startups that have side projects, and their side projects become more successful than their main projects. However, we need more evidences to make side project a norm for software startups. We suggest that it is arguably beneficial to have a side project parallel to the main product development project, taking into consideration of running two projects in parallel.

References

1. Carlson, N.: The real history of Twitter. Business Insider [Online]. http://businessinsider.com/how-twitter-was-founded-2011-4/. Accessed 3 Feb 2016
2. Nazar, J.: 14 Fanous Business Pivots [Online]. http://www.forbes.com/sites/jasonnazar/2013/10/08/14 famous-business-pivots/. Accessed 3 Feb 2016
3. Giardino, C., Bajwa, S., Wang, X., Abrahamsson, P.: Key challenges in early-stage software startups. In: Lassenius, C., Dingsyr, T., Paasivaara, M. (eds.) Agile Processes, in Software Engineering, and Extreme Programming. Volume 212 of Lecture Notes in Business Information Processing, pp. 52–63. Springer International, Cham (2015)
4. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business, New York (2011)
5. Giardino, C., Wang, X., Abrahamsson, P.: Why early-stage software startups fail: a behavioral framework. In: Software Business. Towards Continuous Value Delivery, pp. 21–41. Springer, Cham (2014)
6. Klotins, E., Unterkalmsteiner, M., Gorschek, T.: Software engineering knowledge areas in startup companies: a mapping study. In: Fernandes, J.M., Machado, R.J., Wnuk, K. (eds.) Software Business. Volume 210 of Lecture Notes in Business Information Processing, pp. 245–257. Springer International, Cham (2015)
7. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: a systematic mapping study. Inf. Softw. Technol. **56**(10), 1200–1218 (2014)
8. Bajwa, S.S., Wang, X., Duc, A.N., Abrahamsson, P.: "Failures" to be celebrated: an analysis of major pivots of software startups. Empir. Softw. Eng. Print ISSN (1382-3256) (2016)
9. Bosch, J., Holmstrm Olsson, H., Bjrk, J., Ljungblad, J.: The early stage software startup development model: a framework for operationalizing lean principles in software startups. In: Fitzgerald, B., Conboy, K., Power, K., Valerdi, R., Morgan, L., Stol, K.J. (eds.) Lean Enterprise Software and Systems. Volume 167 of Lecture Notes in Business Information Processing, pp. 1–15. Springer, Berlin (2013)
10. Duc, A.N., Shah, S.M.A., Ambrahamsson, P.: Towards an early stage software startups evolution model. In: 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Limassol, pp. 120–127 (2016)
11. Van der Van, J.S., Bosch, J.: Pivots and architectural decisions: two sides of the same medal? What architecture research and lean startup can learn from each other. In: Proceedings of Eight International Conference on Software Engineering Advances (ICSEA'13), Venice, pp. 310–317 (2013)
12. Berrocal, J., Garcia-Alonso, J., Murillo, J.M.: The fall and rise of nimbees. In: 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Limassol, pp. 136–139 (2016)
13. Nguyen-Duc, A., Seppänen, P., Abrahamsson, P.: Hunter-gatherer cycle: a conceptual model of the evolution of software startups. In: Proceedings of the 2015 International Conference on Software and System Process (ICSSP 2015), pp. 199–203. New York (2015)
14. Terho, H., Suonsyrjä, S., Karisalo, A., Mikkonen, T.: Ways to cross the Rubicon: pivoting in software startups. In: Product-Focused Software Process Improvement Volume 9459 of the Series Lecture Notes in Computer Science (2015)

15. Hirvikoski, K.: Startups pivoting towards value. Data and value-driven software engineering with deep customer insight. In: Münch, J. (ed.) Proceedings of the Seminar No. 58314308, pp. 1–7. University of Helsinki, Helsinki (2014)
16. Münch, J., Fagerholm, F., Johnson, P., Pirttilahti, J., Torkkel, J., Jäarvinen, J.: Creating minimum viable products in industry-academia collaborations. In: Fitzgerald, B., Conboy, K., Power, K., Valerdi, R., Morgan, L., Stol, K.J. (eds.) Lean Enterprise Software and Systems, pp. 137–151. Springer, Berlin (2013)
17. Hokkanen, L., Kuusinen, K., Väänänen, K.: Minimum viable user experience: a framework for supporting product design in startups. In: Sharp, H., Hall, T. (eds.) Agile Processes, in Software Engineering, and Extreme Programming. XP Lecture Notes in Business Information Processing, Vol. 251. Springer, Cham (2016)
18. Hokkanen, L., Väänänen-Vainio-Mattila, K.: UX Work in Startups: Current Practices and Future Needs, Agile Processes in Software Engineering and Extreme Programming (XP), pp. 81–92 (2015)
19. Vartanian, T.P.: Secondary Data Analysis. Oxford University Press, New York (2011)
20. Boslaugh, S.: An introduction to secondary data analysis. In: Secondary Data Sources for Public Health: A Practical Guide. Cambridge University Press, Cambridge (2007)
21. Kitchenham, B.: Guidelines for performing Systematic Literature Reviews in Software Engineering. EBSE Technical Report. EBSE-2007-01. Software Engineering Group, School of Computer Science and Mathematics, Keele University, Keele (2007)
22. Yin, R.: Case Study Research: Design and Methods. Sage, Thousand Oaks, CA (2003)
23. Bajwa, S.S.: Pivoting in Software Startups. PhD thesis, Free University of Bozen – Bolzano, Bozen – Bolzano (2017)

Yes, We Can! Building a Capable Initial Team for a Software Startup

Pertti Seppänen

Abstract Startup companies are based on the founders' innovations and visions of new business opportunities. Software startups are commonly considered as especially innovative. Besides the importance of the innovation and business vision, in the early stages of the startup, the initial team plays a key role in transforming the innovation into a product or a service. At the same time, software startups are often small, immature companies with very limited resources. That highlights the importance of the initial team's capabilities to address the challenges of product development from the innovation—the knowledge, experiences, skills, and other cognitive abilities. In this chapter, we present the results of studies on the initial team's capabilities from the viewpoint of the product development, planning, designing, implementing, and verifying the targeted product or service. The studies were conducted on a group of 13 software startups in Italy, Norway, and Finland. The studies revealed that from a group of very heterogeneous software startups a generic structure of the initial team could be identified, consisting of three different roles, each having a specific set of responsibilities and capability needs. This team structure provides a software startup with a balance between the team's capabilities and problems and challenges to be solved during the early product development process. In addition, we present the sources of the needed capabilities, the initial knowledge, experience, and skills of the founder, and broadening and deepening the initial capabilities by validated learning and by growth toward the identified team structure.

Keywords Software startup · Initial team · Product development · Product development process · Capability needs · Building teams · Learning by doing

P. Seppänen (✉)
M3S/M Group, University of Oulu, Oulu, Finland
e-mail: pertti.seppanen@oulu.fi

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_3

1 Introduction

A software startup's ability to transform innovation to a product or a service is largely affected by the challenges it faces during its early stages, such as time pressure, a small and inexperienced team, dependency on a single product, and general lack of resources [1]. Besides the business value of the innovation, it is crucial that a startup can acquire the knowledge, skills, and capabilities needed to create a product on the innovation.

The existing research on the experiences, skill, and knowledge in case of startups have been focused on the founder and her capabilities [2–4], publishing partly conflicting findings. In the prior studies on the software development work startups, such areas have been addressed, as the life cycle phases [5, 6] and the ways how the startups utilize the established software development methods [7, 8]. Research on the capabilities of the whole team, how the capabilities are acquired, and how they reflect the internal structure and roles of a software startup is, however, scarce. To fill up the research gap, a series of studies were conducted addressing software startups from those less-studied viewpoints.

What is a startup?

- ✓ A startup is an **innovation mission**
- ✓ A startup is a **demanding mission**
- ✓ A startup needs a **team that can realize the innovation to a business case**

2 Prior Research

This section summarizes the prior research into the challenges of software startups and capabilities and roles of their initial teams.

2.1 Prior Research on Software Startup Challenges

A startup faces many challenges from the viewpoint of the initial team's capabilities. The innovation itself may be immature and need changes and adjustments [9, 10]. Recent developments in technology and entirely new user preferences may force rethinking of the business case, sometimes over a longer time [11]. In addition to that, the initial team has many times lacking capabilities in addressing the challenges. The initial team is typically characterized by phenomena that clearly decrease its capabilities [1].

At the same time, the working environment is characterized by phenomena that really make things difficult even for more capable teams [1].

Balancing the above challenges and building a team that is capable in creating viable prototypes and final products out of the innovation is a key success factor of an early software startup.

2.2 Prior Research on Capabilities

The founder lays the basis of the initial team's capabilities [2, 3]. Lots of scientific work have been targeted on the founder, her characteristics, and the ways how she gets and creates her innovations. A characteristic feature of a software startup founder is that she is alert of business opportunities that appear within her sphere of influence or she is able to identify totally new visions for the future [12]. While it is reasonable to assume that a founder's personal capabilities in being alert for opportunities and visionary for future potentials are the best in areas that she knows from the past, studies show that the founder does not need to be a superhero with deep and broad experience on all innovation-related areas [13]. Instead, studies show that the founders may be generalist without being real experts in any areas [4]; they may be just-graduated young people, or managers without relevant technical skills [13].

2.3 Prior Research on Innovation Ownership

A startup is based on an innovation and a vision for a business opportunity. Then, one of the key activities in the startup is testing the feasibility of the innovation and the business vision. Guidance on how to conduct feasibility testing in a systematic way without progressing too long with wrong assumptions and wasting time, work and money is available in the literature. Ries in [9] defines a lean startup thinking focusing on close customer cooperation and objective internal judgment of the possible business value of the innovation. In Early Stage Software Startup Development Model (ESSSDM), Bosch et al. propose a repository of alternative ideas to be tested along the principles of the lean startup [10].

The early stages of a software startup are run by a team that is typically small, possibly inexperienced, and may have various shortages in both material and immaterial resources [1]. What is the ownership of the innovation or innovations in such an initial team? Discussions with practitioners and our recent study [14] indicate that the ownership of the innovation lies typically on the shoulders of a single innovator. She is in most cases also to the founder of the startup and the key driving personality. The situation is similar in startups with several founders—one in the circle of founders has innovated the idea that is brought forward together. To turn her idea into reality, she needs, however, a team that can do the necessary work.

How does such a team look like, and in which way the founder and idea owner can build it?

3 Research Design

This chapter is based on a series of studies we conducted on the same group of real-life software startups utilizing similar qualitative research methods to cover the initial teams of software startups from three viewpoints related to the teams' capabilities, knowledge, and skills. All studies were designed in a similar manner to be able to summarize the findings.

3.1 Case Selection and Research Data Gathering

Studying a group of early-stage software startups provided us with an insight into what development-related capabilities are needed in software startups and gave us a basis for the exploration of how such an initial team is built and structured that has the capabilities necessary to tackle the challenges. Our study was based on a sample group of 13 companies in 4 different geographical areas in Europe having a broad variation in product ideas, customer segments, and utilized technology, as shown in Table 1.

Table 1 Studied software startups

Location	Product type	Customer segment
Italy	WEB application	Photographers, public
Norway	WEB application	Event organizers, public
Norway	WEB/mobile application	Emergency centers, public
Finland, Oulu area	Instrument, embedded software	Specific sector of public
Finland, Oulu area	Instrument, embedded software	Specific sector of public
Finland, Oulu area	IoT device, embedded software	Smart device vendors
Finland, Oulu area	Imaging system, embedded software	Researchers
Finland, Oulu area	WEB application	Nurseries, parents with small children
Finland, Oulu area	Instrument, embedded software	Public
Finland, Oulu area	Software development services	Systems and software companies
Finland, Oulu area	Special IT services	Systems and software companies
Finland, Helsinki area	Aircraft maintenance software	Aircraft carriers
Finland, Helsinki area	Graphical UI platform	Smart device vendors

The research data were collected by conducting semi-structured face-to-face interviews [15] and applying the key informant technique as defined by Marshall [16]. The interviews were held in English, recorded, and transcribed.

3.2 Research Data Analysis

We used thematic synthesis [17, 18] for analyzing the research data qualitatively. The thematic synthesis was conducted by using the NVivo11 tool for coding the research data and combining the codes to higher level themes that summarized the findings related to our research focus, the initial teams, their structures, and means to acquire the necessary capabilities.

4 Results

In this section, we present the results of the thematic syntheses of the research data, revealing fundamental similarities of the initial teams in a group of different software startups.

4.1 Capability Domains for Product Development

Though the details of the development-related capability areas varied between individual software startups, we could identify in our first thematic synthesis a high-level capability mapping with four main areas, as shown in Table 2 [14].

As any other industrial enterprise, a software startup requires a broad variety of capabilities from the personnel of the initial team. For the product development from the innovation, engineering-related capabilities of the initial team are of crucial importance.

Table 2 High-level technical capability domains of software startups

Capability domain	Description
Innovation and application domain	Understanding the requirements and characteristics of the application and the innovation.
Software development domain	Being able to develop functional software fitting to requirements of the application domain and the innovation.
Special technology domain	Being able to develop other functional solutions but software needed in building the product.
Process and quality domain	Being able to conduct the development work in a profitable manner and at an acceptable quality level.

Understanding and mastering the targeted application area is perhaps the most important capability in all industries. The application area defines the main functionality and the key requirements of a new product or service. In many cases, it also sets constraints to the technology basis, architecture, and user experience of the product or service. In a typical case the innovation itself is strongly related to the application—defining the purpose and value of the product for potential customers. The spectrum of application areas tackled by software products and services is huge, varying from scientific applications and life-maintaining instruments to communications, Internet of Things, entertainment, games, and further to toys or similar simple products. Knowing the constraints in terms of technologies, cost structures, and customer preferences helps a startup to plan and allocate its investments in a reasonable way.

In software startups, the main implementation technology is software. Thus, software development capabilities are an absolute necessity, the initial team must be able to turn the innovation to a working software product or service. During the last decades, the art of developing software, software engineering, has undergone big development steps driven by both hard factors, such as technology developments, new application areas, or business constraints, and by soft factors, such as user preferences and even fashions. A software startup can select from a broad palette of software development methods, platforms, design practices, programming languages, operating systems, testing tools, and many other solutions built to support application development. Simplified, however, the initial team must be able to plan, design, implement, and verify functional solutions profitably and at the quality level typical for the application area—it must master relevant software development processes.

In some application areas, the product or service requires other implementation technologies but software. That is typical for embedded products where the software functionality requires development supporting hardware or even mechanics.

4.2 Team Roles

In the second thematic synthesis round, we were able to identify three different personnel roles, each contributing to the overall capability level of the startups in a different way. The personnel roles are shown in Table 3 [14].

Table 3 Team roles

Role	Description
Founder	The founder or the leading person of multi-founder group being the owner of the innovation.
Expert	A qualified professional with strong competencies in specific area(s).
Developer	Member of the development team focusing on low-level design, implementation, and testing task.

The founders were the key persons of the studied startups. Even in cases with several founders, e.g., initial shareholders, there was one person who had come up with the innovation or product idea and was the main person in bringing it forward. She is referred in the following as “the founder.” Even though several founders of the studied startups had prior experience in developing software and managing software development teams, the additional workforce was needed in the actual product development work.

From the head count perspective, most of the initial team members were software developers, people hired to the initial team with a clear focus in designing, developing, and testing the software to the new product. The background of the developers in our study varied a lot. One main division line was between the developers got from service-providing software houses as subcontractors and the developers being hired as personnel of the startups.

Deploying subcontractors for the development work was justified by several reasons, the most common of which was avoiding economic risks that may appear in hiring their own personnel. Even though the unit costs of subcontracted personnel may be in short term somewhat higher than those of hired personnel, subcontracting was seen less risky over a longer run, because legal employer obligations fall on the side of the service providing company. Another reason for subcontracting was also that a startup gathered in that way highly qualified developers to the development team, mitigating the risks of poor software quality or slippages in time schedules. In some cases, subcontracting was also used for getting capabilities in specific technology areas.

The experiences and skills of the hired developers, in turn, varied a lot in our study group. Being young organizations and having limited resources, the startups were forced to balance between the requirements set by the targeted products and the costs of developers with longer experiences in the field. The balancing between the needs and the economic possibilities was taken care of in different ways. Some companies hired only students who were assumed to be cheaper and more willing to work in startup-type companies, some invested in fewer but highly qualified professionals. In the former case, the founder herself had a solid experience and knowledge base on software development and was therefore confident in being able to lead the work of less experienced developers. In the latter case, the reasons to hire experienced developers were, for instance, the focus on a specific technology or especially high-quality requirements. Typical solutions were to hire old workmates of the founder, or persons with otherwise known professional careers.

Even through the above explained way of hiring professionals to the positions of developers due to specific demands, we could identify, besides the founder and the development team, an additional personnel role in the software startups—the experts. Experts were people who compensated for the capability shortages of rest of the team, especially the shortages of the founder. Mostly the experts’ contributions were focused on the areas of the special technology domain or the process and quality domain (Table 2), but in cases of founders without personal software development experiences the experts also compensated the founders’ missing software capabilities.

Experts worked both as hired personnel and as subcontractors. Subcontracted experts were used especially in solving the problems of the specific technology areas needed in the product development [19]. Such needs appeared due to the fact that even experienced founders were sometimes less familiar with all the required technologies because experienced founders tended to have new or more ambitious product views. In those cases, the experts' contributions were very focused, both in terms of problems and time.

Experts were hired to the company in cases of long-lasting needs to compensate for the capability shortages of the founder, and the focus of the contribution varied from some specific areas to the whole product development. In two cases with just-graduated founders without own software development experiences, the key contribution of the experts was to build a capable software development team—after failed attempts of the founders.

In the early stages, startups might experience different contractors for developing prototypes [19]. Outsourced tasks or sub-modules at these phases are small scale and experimental. Given that, startup founders who lack technical competence often choose to outsource as a shortcut to a later stage of startups, where they can attract funding for proper product development. In some other cases, startup looks for a sustainable strategy for product development, using their unique advantages, such as a personal relationship with a reliable outsourcing team, or successful collaboration previously [19].

4.3 Means to Build the Team Capabilities

The third round of thematic synthesis was conducted to identify the means to acquire the capabilities in the initial team of a software startup. Though the details of how the teams were built up in the case startups varied, three high-level means that were common for all cases were identified, as presented in Table 4 [13].

Table 4 Means to acquire team capabilities

Acquiring means	Description
Founders' initial capabilities	The initial experiences, knowledge, skills, and competences of the founder(s).
Additional capabilities through team expansion	The experiences, knowledge, skills, and competences of new team members.
Additional capabilities through team growth	The experiences, knowledge, skills, and competences gained in a learning-by-doing manner during the actual development work.

5 Discussion

In this section, we summarize and discuss the findings of our studies. As the focus of this chapter is the building of a capable initial team for a software startup, we first explore the findings from the viewpoint of our third thematic synthesis, the means to acquire the team's capabilities. Then we summarize all three viewpoints and present a schematic model for the capability structure of a software startup's initial team, including the capability areas, team roles, and means to acquire capabilities.

5.1 The Founder: The Initial Capabilities

Our findings are along the results of the prior research—the founder builds both the innovation and the initial team's capabilities on her personal capabilities, knowledge, and experiences. A founder's prior experiences and accumulated domain knowledge lay the basis for both a successful business case and a smooth development of the product or service [2, 20]. Examples of experienced founders are experts who have worked earlier in another, possibly bigger, company on the same business and technology areas, serial entrepreneurs, or persons gaining their knowledge through personal interests, such as contributions to open-source projects [13]. In those cases, the founder's own capabilities build a strong basis for the startup, the founder masters the key areas of her new enterprise, and the rest of the initial team is built typically to broaden the development-related capacity of the startup.

Cases of the opposite—founders without prior experience—are young people who vote for entrepreneurship right after graduation, people who change their interest and future plans to some totally new area not known in advance, or people who master the needed capability areas only partially, such as managers without own software development skills [13]. In those cases, the founder needs a team that is capable of compensating for her shortages, whether they are in the application domain, technology domain, business domain, or any necessary work domain in the early stages of the company [21].

5.2 Team Mates: Capabilities Through Growth

The basic means to broaden and deepen the capabilities of a software startup is through growth—gathering the initial team to carry out the development work together with the founder. Because of limited resources the initial team is typically small [1]. Thus, it is crucially important for the founder and the startup to ensure that the team is in balance with the challenges faced during the development of the product or service.

Several internal and external constraints affect the building of the initial team, such as the innovation itself, the needed technology, the availability of qualified workforces, the economic resources of the startup, and also the founder's skills to identify the needs and find right people.

An experienced founder has several benefits compared to less experienced founders. She has a better chance for networking with other professionals, among whom she may find applicants willing to join the initial team. The network may consist of old workmates, subcontractors, or subordinates from different phases of the founder's career, or people she had learned to know in other professional circumstances. Such networks are highly valuable for a founder from many viewpoints of a software startup, getting founding, identifying potential customers, building different ecosystems, and building a team of her own.

Another important benefit is the ability of an experienced founder to estimate more objectively the requirements set by the innovation and the technologies used to realize it. Understanding the need is a prerequisite for building a balanced team that is able to carry out the development team but is not wasting the economic resources of the startup.

A software startup may end up in a situation opposite to the above if the founder is not competent enough to foresee the future challenges realistically and to evaluate the software development skills of the applicants she is going to hire. A crude mistake that may lead to a total failure in developing the product or service, laying off the first team, gathering a new team, and starting the work from the beginning if the resources allow [22]. If the founder ends up in such a situation, the most crucial step is to find an experienced software professional, carry out reasonable introduction to her, ensure her commitment toward the startup, and let her take care of gathering a new team [22].

A similar approach is to be recommended if the implementation of the innovation requires, especially difficult or unfamiliar technical solutions or sets other strict requirements, such as very high quality or reliability. Even an experienced founder may face such a situation if the innovation is highly ambitious or is outside of her prior experience areas.

Building the initial team of a software startup is risky for both the founder and for her potential team members. A startup is typically a new organization with limited resources [1], and gathering the initial team means many times selling only ideas and visions, seldom real existing benefits. That leads many times to approaches different from more established companies.

5.3 Piloting the Implementations: Capabilities Through Learning

The recent approaches of product and software development, incremental, agile and lean development practices, and developments in both hardware and software platforms have lowered the technology-related entry barriers of software startups.

Table 5 Internal challenges for the initial team

Characteristic	Description
Lack of resources	Economical, human, and physical resources are extremely limited.
Third-party dependency	Due to lack of resources, to build their product, startups rely heavily on external solutions: External APIs, open-source software, outsourcing, commercial off-the-shelf solutions, etc.
Small team	Startups start with a small number of individuals.
Low-experienced team	A good part of the development team is formed by people with less than 5 years of experience and often recent graduates.
New company	The company has been recently created.
Little work history	The basis of an organizational culture is not present initially.

Table 6 External challenges for the initial team

Characteristic	Description
High reactiveness need	Startups should be able to quickly react to changes of the underlying market, technologies, and products (compared to more established companies).
Innovation need	Given the highly competitive ecosystem, startups need to focus on innovative segments of the market.
Uncertainty	Startups deal with highly uncertain ecosystems from many perspectives: market, product, competition, people, and finance.
Rapidly evolving	Successful startups aim to grow rapidly.
Time pressure	The environment often forces startups to release fast and to work under constant pressure (terms sheets, demo days, and investors' requests).
One product	A company's activities gravitate around one product/service only.
Highly risky	The failure rate of startups is extremely high.
Not self-sustaining	Especially in the early stages, startups need external funding to sustain their activities (venture capitalist, angel investments, personal funds, etc.).

The principles of the lean startup crystallize an iterative, incremental development of minimum viable products (MVPs) that are functional solutions providing the key functionalities and characteristics of the targeted product with a minimum development effort. The feasibility of MVPs is measured with the actual customers, fitting well to software startups, developing new, innovative products, or services in a small team.

A development process following the principles of the lean startup provides a software startup with several benefits that help the startup to tackle the challenges listed in Tables 5 and 6. Although the main focus is avoiding wasted development efforts, other benefits can be seen:

1. Iterative and incremental development is easier for small teams.
2. Iterative and incremental development creates faster tangible results that provide the team with faster learning.
3. Close customer cooperation provides the team with right learning, guiding the team faster to right direction.

Thus, it is no surprise that all product-developing startups in our study group utilized iterative and incremental development approaches, though not necessarily following any fixed methodology, like the lean startup, by the book. Also, close customer cooperation was utilized in the cases where the potential customers were accessible and willing to cooperate.

An interesting phenomenon was identified, laying the basis for future learning—copying from other existing products. In our sample of 11 product-developing startups, 7 developed a variant or direct competitor of some existing product, 2 had fully new innovations, and 2 developed combinations of existing products and new innovations [14]. Both service-offering startups continued the businesses the founders had carried out in another, more established companies. In all copying cases, the business vision was, however, new. It varied from utilizing new improved technology to developing similar products for different markets. Copying from existing products not only decreased overall risks and made ensuring the business feasibility easier but was a source of valuable learning. In addition, copying made it easier for the founder to figure out the capabilities needed for the implementation and, thus, to build a balanced initial team.

Independently of the product type or customer segment, the studied startups were utilizing the learning from their iterative and incremental development processes for improving the capabilities of the initial team, some practices to be mentioned are:

1. Technical feasibility studies in very early stages of the startup or even before founding the company.
2. Utilizing external experts for helping to solve, especially difficult technical problems.
3. Developing minimum viable products based on fast and easy-to-use technology platforms, different from the one of the final solutions.
4. Developing series of prototypes with stepwise increasing functionality.
5. Having close cooperation with the customers.

The founder's initial skills and knowledge lay the basis of the capability structure of a software startup and rational growth brings new capabilities to the initial team. In both cases, the capabilities come from the past of the individuals, being mostly experiences and learning from other situations and other environments, not necessarily directly applicable in the context of the new startup.

The capability increments gained by learning during the iterative and incremental development process are, in turn, problem specific, company specific, and customer specific. Thus, it is reasonable to claim that such capabilities are more valuable for the startup than the ones from the past. The capabilities acquired in the actual product development context provide a software startup with a resource base that is rare, difficult to imitate, or difficult to substitute, which in terms of the resource-based view gives the startup a sustainable competitive advantage [23].

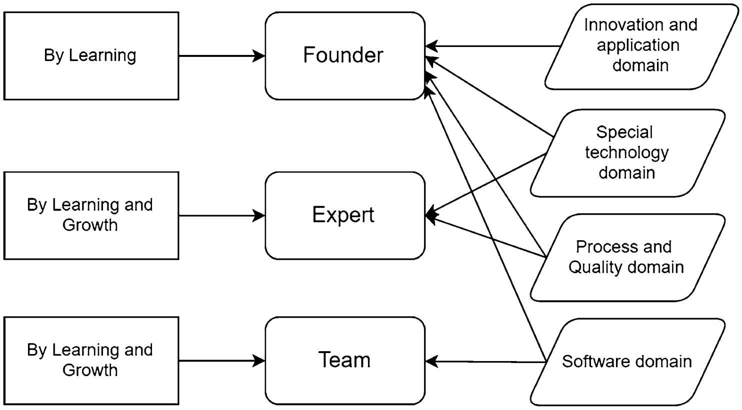

Fig. 1 Capability structure of a software startup's initial team

5.4 Capability Structure of the Initial Team

Out of the findings, we could figure out a capability structure of a software startup's initial team that combines the capability areas, the team roles, and the means of capability acquiring identified in our studies. The structure is presented in Fig. 1 [22].

The structure highlights differences both between the roles and between the capability domains. In practical situations, the borderlines between the roles may not be that clear. The same person can play different roles in different situations or at different times, or an expert can master different capability domains. Similarly, the capability domains overlap each other, for instance, the software capability domain is strongly interlinked both to the innovation, application, process, and quality domains.

6 Conclusions

The findings of our study on the initial teams of software startups open for the founders of new startups several viewpoints helping them in figuring out the processes from an innovation to a product and the team that is needed in carrying out the work. Also, a just-graduated student can utilize some findings when considering whether to apply for a job in a software startup or to accept a position offer from one.

In a capable software startup, there must be an innovation owner, a person with confidence to the innovation, and willingness to bring it further to a product. From our study group's perspective that seems to be self-evident, because all founders were committed to the innovation and to the future product. It is known, however, that this is not always the case, but a startup may need to struggle in finding out an idea to bring further [9, 10]. From a just-graduated student's point of view, a committed idea owner is of crucial importance—one should avoid employments in software startups without a reasonably strong commitment of bringing the innovation to a product. Even though the founder's or the team's commitment does not guarantee any business success for the developed product, it provides a new comer with a clearer focus and a more stable direction to the product development. That, in turn, offers her better learning points on how the development work in software startups is and how she could utilize those learning in future career.

Our findings also indicate that not all members of the initial team need to be especially innovative. Most of the individuals of the team are focusing on software development duties. Especially when the founder or the hired expert is a talented software professional, a software startup may offer a just-graduated an environment where she can practice many relevant disciplines. Bigger companies may be organized in silos for disciplines, such as requirements engineering, coding, and testing, offering only a narrow base for learning by doing.

For a new founder, the key finding is that one does not need to be a superman in order to build a software startup. Missing knowledge can be gained by orienting deeply on the innovation and problem domains—even by studying existing innovations and products, and the actual shortages in capabilities can be compensated by careful building of the initial team.

To build a successful team, the founder must be able to evaluate the future challenges and their own shortages in an objective manner. Identifying own weak areas and looking for capable teammates is one of the most important issues when moving from an idea-level innovation to a severe product development. Though not directly addressed in our study, it is reasonable to assume that the funding bodies carefully evaluate not only the innovation but also the team that has been built to realize the innovation.

Both prior studies [9, 10] and our findings show that not all challenges need to be tackled before founding a startup. Utilizing iterative and incremental development approaches, the founder and the whole team can acquire improved capabilities that are especially valuable because they are based on learning from the actual development process.

Main findings

- ✓ A team that can realize the innovation has three key roles: the **founder**, the **expert**, and the **team member**
- ✓ The founder owns the innovation
- ✓ The expert compensates for the founder's capability shortages
- ✓ The team member focuses on implementation-related tasks
- ✓ A startup is a **learning-by-doing mission**

References

1. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: a systematic mapping study. Inf. Softw. Technol. **56**, 1200–1218 (2014)
2. Bosma, N., Van Praag, M., Thurik, R., De Wit, G.: The value of human and social capital investments for the business performance of startups. Small Bus. Econ. **23**, 227–236 (2004)
3. Marvel, M.R., Lumpkin, G.T.: Technology entrepreneurs' human capital and its effects on innovation radicalness. Entrep. Theory Pract. **31**, 807–828 (2007)
4. Lazear, E.P.: Balanced skills and entrepreneurship. Am. Econ. Rev. **94**, 208–211 (2004)
5. Blank, S.: The Four Steps to the Epiphany: Successful Strategies for Startups that Win. K&S Ranch, Menlo Park, CA (2013)
6. Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: the greenfield startup model. IEEE Trans. Softw. Eng. **42**, 585–604 (2016)
7. Giardino, C., Unterkalmsteiner, M., Paternoster, N., Gorschek, T., Abrahamsson, P.: What do we know about software development in startups? Software IEEE. **31**, 28–32 (2014)
8. Klotins, E., Unterkalmsteiner, M., Gorschek, T.: Software Engineering Knowledge Areas in Startup Companies: A Mapping Study. Presented at the (2015)
9. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Publishing Group, New York (2011)
10. Bosch, J., Olsson Holmström, H., Björk, J., Ljungblad, J.: The early stage software startup development model: a framework for operationalizing lean principles in software startups. In: Lean Enterprise Software and Systems, pp. 1–15. Springer, Berlin (2013)
11. Ojala, A.: Business models and opportunity creation: how IT entrepreneurs create and develop business models under uncertainty. Inf. Syst. J. **26**, 451–476 (2016)
12. Alvarez, S.A., Barney, J.B.: Discovery and creation: alternative theories of entrepreneurial action. Strategic Entrep. **26**, 11–26 (2007)
13. Seppänen, P., Liukkunen, K., Oivo, M.: Little big team: acquiring human capital in software startups. In: International Conference on Product-Focused Software Process Improvement, pp. 280–296 (2017)
14. Seppänen, P., Oivo, M., Liukkunen, K.: The initial team of a software startup, narrow-shouldered innovation and broad-shouldered implementation. In: To Be Published in 22nd ICE/IEEE International Technology Management Conference (2016)
15. Lethbridge, T.C., Sim, S.E., Singer, J.: Studying software engineers: data collection techniques for software field studies. Empir. Softw. Eng. **10**, 311–341 (2005)
16. Marshall, M.N.: The key informant technique. Fam. Pract. **13**, 92–97 (1996)
17. Cruzes, D.S., Dybå, T., Runeson, P., Höst, M.: Case studies synthesis: a thematic, cross-case, and narrative synthesis worked example. Empir. Softw. Eng. **20**, 1634–1665 (2015)
18. Cruzes, D.S., Dybå, T.: Recommended steps for thematic synthesis in software engineering. In: International Symposium on Empirical Software Engineering and Measurement, pp. 275–284. IEEE, Piscataway (2011)
19. Duc, A.N., Abrahamsson, P.: Exploring the outsourcing relationship in software startups: a multiple case study. In: Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, pp. 134–143. ACM, New York (2017)
20. Seppänen, P., Tripathi, N., Oivo, M., Liukkunen, K.: How are product ideas validated? In: International Conference of Software Business, pp. 3–17 (2017)
21. Nguyen-Duc, A., Seppänen, P., Abrahamsson, P.: Hunter-gatherer cycle: a conceptual model of the evolution of software startups. In: Proceedings of the 2015 International Conference on Software and System Process, pp. 199–203. ACM, New York (2015)
22. Seppänen, P.: Balanced Initial Teams in Early-Stage Software Startups: Building a Team Fitting to the Problems and Challenges. (Doctoral Dissertation, University of Oulu, Finland) (2018)
23. Barney, J.B.: Firm resources and sustained competitive advantage. J. Manag. **17**, 99–120 (1991)

The Perception and Management of Technical Debt in Software Startups

Cecilia Apa, Helvio Jeronimo, Luciana M. Nascimento, Diego Vallespir, and Guilherme Horta Travassos

Abstract The software startups are a particular scenario where Technical Debt (TD) may occur in an intentional or unintentional way. However, the current knowledge about the perception and management of TD are mainly related to mature software organizations. This chapter contextualizes the startups' characteristics that can lead to TD incurrence, the concepts related to TD and its management, and presents the results of a survey with Uruguayan software startups. The survey's primary goal was to understand how software startups perceive and manage TD in their projects. The results refer to the level of understanding of the startup's practitioners concerning TD concept, the adopted Technical Debt Management (TDM) activities, and the strategies and technologies used in their projects to support such activities. The findings show that startups seem to invest time and effort in TDM activities being TD prevention, one of the most conducted activities. Besides that, it was observed that the participant's experience and the size of the organization seem to be also related to the perception and management of TD.

Keywords Technical Debt · Management · Startup · Survey · Empirical software engineering

1 Introduction

In the last years, Technical Debt (TD) has been an exciting research topic for the software industry and academia. TD is associated with technical decisions in the software development that can bring benefits in the short term, like savings of time (schedule) and cost reduction. However, in the long term, these decisions may bring

C. Apa (✉) · D. Vallespir
Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay
e-mail: ceapa@fing.edu.uy; dvallesp@fing.edu.uy

H. Jeronimo · L. M. Nascimento · G. H. Travassos
Universidade Federal do Rio de Janeiro/UFRJ - COPPE, Rio de Janeiro, Brasil
e-mail: jeronimohjr@cos.ufrj.br; lmnascimento@cos.ufrj.br; ght@cos.ufrj.br

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_4

some risks to internal software quality, hindering the maintenance and evolution of software products. As far as it can be observed, most, whether not all, software projects face TD [1, 2].

Software startups are a particular and exciting context where TD may emerge. At their initial stages (ideation and creation), TD is rarely considered. From the moment the ideas materialize in software products with a high probability of success, TD makes sense and starts to be observed. At this point, the decisions taken can bring risks to evolve the product, project, and/or business. Gralha et al. [3] point out that TD is a dimension needed to take into account to support the requirements practice evolution in the software startup. Even though TD has been a topic of interest of many studies in the last years, most research on TD focuses on mature software organizations [4, 5]. Therefore, there is a gap regarding the perception and management of TD in software startups.

An exploratory study with startups in Brazil [6] indicates that TD emerges after its initial stages due to their contextual characteristics. Other previous studies reported similar behaviors [5, 7–10]. After the initial stages, the management of such debts is claimed as essential in the software development life cycle. Furthermore, the perception of software product quality tends to change over time. In the initial stages, characteristics such as usability and functional suitability are considered significant since the main goals are related to the acceptance and success of the software product in the market. In the next stage, when the product has a high probability of success, or when changes occur in the team, or the number of clients/users increases, the quality concerns are associated with other characteristics such as maintainability and evolvability to meet the required changes and product scalability. Once the perception of software products evolves in startup organizations, their adopted software engineering practices also need to evolve. Therefore, issues related to internal software quality may be treated to reduce the risks of TD aiming to keep a proper stabilization, evolution, and maturation of the software product throughout its life cycle. Thereby, the first step is to understand how startups perceive and manage TD in their projects.

Recently, we surveyed practitioners engaged in Uruguayan software startup organizations to know how such individuals perceive and manage TD in their projects. This chapter presents the results of this survey, such as the level of understanding of the practitioners concerning TD concept, the adopted Technical Debt Management (TDM) activities, and the strategies and technologies used in their projects to support each one of TDM activities. The results and discussions presented in the following sections intend to contribute to both industry and academia on the perception, management, and further empirical studies regarding TD.

The chapter is organized into five sections. Section 2 presents the fundamental concepts. Section 3 describes the results of the conducted study. Next, Sect. 4 presents discussions about the main findings of the study. Finally, Sect. 5 addresses the main findings and the relevance of the given evidence considering the perspectives of industry and academia.

2 Fundamental Concepts

This section presents characterization of TD in a software project (Sect. 2.1) and the startup's contextual characteristics emphasizing some factors that can lead to the incurrence of TD (Sect. 2.2).

2.1 Technical Debt and Its Management

Ward Cunningham [11] coined the "Technical Debt" term (TD) as a metaphor in the software industry when discussing with stakeholders the consequences of releasing a poorly written piece of code to accelerate the development process. Since then, this term has been used as a metaphor to refer to technical issues occurring during software development as well as to allow better communication with nontechnical stakeholders. Project managers and executives across the industry have adopted this metaphor because it describes occurrences during the software life cycle in a language that industry understands [2, 12].

Initially, TD was used continuously to refer to the issues associated with the quality of source code. It was due to the shortcuts and workarounds took during software development to meet the constant demands, in which these issues may affect the quality of software products and their maintenance activities in the future. Currently, studies (presenting a more consolidated perspective motivation to the Software Engineering community) toward the understanding and definition of this phenomenon state that TD is: "*a collection of design or implementation constructs that are expedient in the short term but set up a technical context that can make future changes more costly or impossible. Technical debt presents an actual or contingent liability whose impact is limited to internal system qualities, primarily maintainability and evolvability*" [12].

In a broader scope, it is possible to observe that TD can be associated with intentional or unintentional technical decisions regarding shortcuts and workarounds taken in software development. As argued by Becker et al. [13], TD decisions involve intertemporal choices, i.e., decisions involving trade-offs among costs and benefits occurring at different times. Such decisions are taken in different situations, being motivated by technical or business factors.

Besides that, the presented definition shows that TD may occur in distinct software development phases, and it may be associated with different software artifacts. Therefore, different TD types can be incurred in the software development process, and each one of them may influence differently.

Table 1 summarizes the TD types and their descriptions with some examples. The TD types are based on the classification presented in Li et al. [14] and Alves et al. [15]. As shown in Table 1, usability, defect, people, infrastructure, and process also are mentioned as types of TD. However, TD cannot be generalized to all the software issues currently faced in the projects, since it is limited to internal quality

Table 1 TD types

TD type	Description
Architecture Debt	Refers to suboptimal architecture solutions, impacting the internal quality (e.g., violations of the adequate and adopted architectural)
Build Debt	Refers to issues in the build process that may make this process harder (e.g., files of build containing code source that does not add value to this task and software products and manual build)
Code Debt	Refers to the issues in the source code that may hamper the modularity, reusability, analyzability, and modifiability of the software products (e.g., code source that does not meet the required coding standards)
Documentation Debt	Refers to the issues found in the documentation (e.g., lack, insufficient, outdated, or inadequate documentation of the artifacts' software)
Design Debt	Refers to technical shortcuts taken in the detailed design and may be found by analyzing the source code or design models (e.g., violations of the principles of good object-oriented design, code smells, and grimes)
Requirement Debt	Refers to the distance between the optimal requirements that need to be implemented and the actual software products implementation, under domain assumptions and constraints (e.g., an implemented requirement which, in a way, does not fully satisfy all the nonfunctional requirements)
Service Debt	Refers to inadequate use of software services (e.g., poor selection and use of software services). Services refer to independent technologies that offer specific business functionality. These are described in a standardized way having published interfaces and communicating with other services through remote calls
Test Debt	Refers to issues related to testing activities (e.g., lack, insufficient or inadequate tests; low tests coverage; and deferring testing)
Versioning Debt	Refers to issues related to source code versioning (e.g., unnecessary code forks)
Usability Debt	Refers to inappropriate decisions related to software usability that will need to be adjusted later (e.g., lack, insufficient, outdated, or inadequate usability standards)
Defect Debt	Refers to the known and deferred defects (e.g., postponed decisions on fix defects)
People Debt	Refers to issues related to people leading to the incurrence of TD (e.g., lack of knowledge and negligent attitudes)
Infrastructure Debt	Refers to issues related to the infrastructure of development or operation that may contribute to the incurrence of TD (e.g., lack, insufficient, outdated, or inadequate tools or components to support the activities of development, deployment, and operation)
Process Debt	Refers to issues related to the adopted software process that may contribute to the incurrence of TD (e.g., an inadequate process not providing proper support to the development activities)

issues. Then, issues related to usability and defect should not be considered types of TD because they are associated with the external quality characteristics of software products. Besides, when analyzing the technical literature, it is possible to identify that people, infrastructure, and process refer to contributory factors to TD incurrence [1, 2, 16].

Technical debts in software projects
✓ It may bring some benefits and risks to the software project ecosystem and the quality of their software products.
✓ Its occurrence is motivated by factors associated with business, team members, and/or technical aspects.
✓ Different TD types can occur in distinct software development phases.
✓ Each TD type may influence differently.

As it has been said before, TD is always inevitable in the software development scenario [1, 2]. For example, many decisions must be taken when a project software starts, but it is hard for the team to have a complete understanding of the problem as a whole, which turns inevitable the incurrence of unintentional TD. On the other hand, TD may be intentionally incurred to achieve some business advantages by sacrificing the internal quality in the short term. Thus, TD can influence positively (intentional and managed) or negatively (unintentional and not managed) to the software project ecosystem and the quality of their software products. When TD is managed and perceived in software projects, it has the potential to support the delivery of value to customers in the short term. On the other hand, in the long term, the maintenance and evolution of software products can be hampered when the debts are not known nor managed in the projects. Then, TD is not necessarily a "bad thing" if it is perceived and managed strategically in the software project.

Therefore, the perception and management of TD should be a continuous activity in software projects aiming to handle the trade-offs associated with the constant demands for software deliveries, and the risks it poses to the internal software quality. Technical Debt Management (TDM) in software projects consists of the following activities: identification, measurement, prioritization, prevention, monitoring, repayment, and representation/documentation [14, 15, 17].

In the TD technical literature, it is possible to identify proposals of software technologies (i.e., practices, methods, techniques, processes, and tools) to support the TDM activities regarding different types of TD. Besides that, there are investigations covering the perception of TD and its management under the software industry point of view [18–24].

A survey was conducted with regular Brazilian Software Organizations (BSOs) in 2018 regarding the perception of TD and its management in software organizations [24]. The primary goal of this study was to acquire knowledge on the perception of TD metaphor and its management in the industry (i.e., the adopted strategies, activities, and technologies), using the engaged practitioners as proxies. This survey was performed with 58 practitioners, representing about 12 organizations and 30 software projects. Concerning the TD awareness, the results of this study indicated that Brazilian software practitioners did not reach any consensus on how they perceive TD, and TD is still unknown to a considerable fraction of the participants. It was observed that 50% of the participants consider that TD should also be associated with external quality issues, indicating a misconception of what should be considered TD, because they also associated TD with any issue occurring during the software development.

On the other hand, it was also observed some agreement among the respondents on associating TD to internal quality issues, in which it presents an alignment with the TD definition indicated in the technical literature [2].

Regarding the perception of TD, it was identified that a low number of respondents (42%) informed to perceive TD in their software projects.

Regarding the management of TD, the results of this survey indicate that a few respondents (25%) indicated to adopt TD management activities in their projects, with no consensus on which TDM activities are more relevant to the surveyed BSOs. However, the prevention of TD was considered relevant by half of the participants that answered the question related to the importance of TDM activities. On the other hand, despite the number of participants indicating the importance of the prevention of TD, only two respondents (two BSOs) reported performing activities of prevention of TD.

The existence of types of TD and its management is dependent on the software project context. In this case, the project context refers to the practices of software engineering and software artifacts that are required during the software development life cycle. Therefore, TD's perception and its management can be influenced by the project context. A set of tools and software technologies that can be used to support TDM activities in software projects is presented in [14, 24]. However, further studies looking for evidence on their effectiveness and efficiency must be performed, including scenarios (from where TD may emerge), such as software startups [7]. As reported by Besker et al. [5], the intentional TD may be considered as a useful strategy in a short time to balance the benefits of time-to-market and reduced resources in software startups. However, unmanaged TD may bring negative consequences [9]. Therefore, software engineering practices must be preemptively adopted to support the perception and management of TD, aiming to address the risks that it imposes to this scenario.

The perception of TD and its management

- ✓ The perception and management of TD should be a continuous activity in software projects.
- ✓ Studies report that the perception and management of TD are still incipient in software organizations.
- ✓ The software project context can influence how TD may be perceived and managed.

2.2 Software Startup Context

Startup organizations aim at developing innovative solutions to unmet markets by using or generating cutting-edge technologies while operating under highly uncertain conditions, with a severe lack of resources, having little working history, and in an environment that is highly dynamic due to influences of changes in market and technologies [25]. Despite the lack of consensus about what characterizes a startup, authors usually report as characteristics of startups the need to develop software fast and achieve a short time-to-deliver, focus on the product instead of the project, and the adoption of methods that allow reducing the uncertainties gradually [8].

The software development at startups is market driven, which means that requirements are invented [26]. The lack of customers specifying demands, validating, and paying for their development, contributes to increasing the uncertainties regarding the problem and the solution fit. It is only possible to validate the requirements by engaging representatives of the target customers to evaluate prototypes, or by launching software versions in the market. The company search for a scalable business model also contributes to elevating such uncertainties, since the startup lacks knowledge about the real needs and preferences of the target customers, how to reach them, how much they would pay for the proposed solution, besides other concerns.

In general, startups adopt incremental and evolutionary approaches as development methods so that they can perform validated learning cycles [27, 28], also known as probing and learning cycles [29], or continuous experimentation [30]. In each cycle, the startup develops a Minimum Viable Product (MVP) to support collecting data about and analyzing a subset of hypothesis regarding the product ideas and/or business model definitions. Grounded by data, the startup can decide to keep the product features and proceed to develop the next increment, to evolve or change them (pivot) in the subsequent validated learning cycle, or decide to give up the endeavor.

In this context, the adoption of some Software Engineering practices is reported as inadequate, mainly in the initial development cycles [27]. Startups rely on tacit knowledge, giving up documentation, and adopting simple practices of engineering and product management. Some of the fundamental characteristics and practices of startups support it, such as small and collocated teams, frequent face-to-face communication, use of public standards or frameworks, third-party solutions, and extensive use of prototypes [8]. However, if a startup can survive, the development practices taken as strategic in the early stages may pose risks in the next stages of not achieving the speed and qualities desired in the product evolution [10].

Startups that survive evolve from conception to fruitful company context in stages, setting up different development contexts and challenges. Startup's life cycle models such as the Customer Development model [28]—which was made accessible as part of the Lean Startup method [31]—and the Crowne's model [32] divide the startup evolution stages in relation to the level of the business model and/or product uncertainties and the startup market entrance. These models emphasize that practices adopted may change to support product development, according to the challenges that arise in each stage.

Concerning software quality, Technical Debt issues have gained the attention of researchers [33]. The decisions taken regarding the adoption of development practices in the early stages help startups to be fast. However, as observed by

Giardino et al. [10], they lead to accumulation of TD that hinders the productivity in later stages and may influence the internal software quality. Observing the evolution of 16 startups, Gralha et al. [3] propose a theory about requirements practice evolution in startups, including TD when dealing with new requirements. According to the theory, startups evolve their TD practices from the *Known* and *Accepted* stage to *Tracked* and *Recorded* one before they reach a stage where TD is managed and controlled. The authors also observed that the number of employees, the number of the software features, the clients' retention rate, and the occurrence of negative feedback from clients might cause the startup to change its TDM practices about requirements.

As the company grows, new people join the team. In this scenario, the lack of registered information about the project and the software characteristics lower the team productivity and heighten the chances of inserting defects in the software, since it is hard to pass the accumulated tacit knowledge to the new hires [10]. The high level of tacit knowledge can generate discrepancies between the developers' mental models on the software behavior and its real behavior, or the situation in which no one remembers why a software component behaves as it does [34]. These situations can lead developers to take wrong decisions when evolving the software features.

When the number of users increases, the customer requests also increase, and the requirements and issues to solve may become unmanageable. The demands about product quality become higher, raising the risk of losing customers. As observed by Nascimento and Travassos [6], the costs of quality assurance activities without the support of documentation about the software features and test cases may turn high.

Even in the early stages, some contextual characteristics of startups that support their practices can change and, as a result, give rise to TD. In [6], it is reported that a startup can hire inexperienced people when exploring the first ideas to a software product and, once it is more confident in the product chance of success, the startup may decide to hire more experienced developers and then it needs to document the current software features in order to integrate people on the team. Crowne [32] also reports that, yet in the Startup early stage, the product platform may become unrecognizable for the team when the importance of the product's technologies and/or components is not discussed nor managed.

As discussed, initially, some practices of software engineering tend to be avoided because an initial solution of a software product is delivered as an MVP to experiment with the ideas developed for the product and give adequate feedback about the new features. However, this initial approach may bring some risks to internal software quality, for example, affecting the evolvability of such products in the future. Therefore, the TD perception and the evolution of practices in startups to manage it may be essential to reduce the risks in the software product, project, and/or business.

However, the current knowledge about the perception and management of TD in the software startup context still requires investigations. The next section presents an exploratory study, in which it is possible to observe some perceptions of TD and its management at the software startups context.

TD and its management in software startups
✓ The startup's contextual characteristics can lead to intentional or unintentional occurrence of TD in this scenario.
✓ In the initial stages of software startups, TD can be considered as an investment (strategic).
✓ The management of TD is relevant as early as internal software quality issues are handled to keep a suitable stabilization, maintenance, and evolution of such products.

3 The Perception of TD and Its Management at Software Startups' Context

Between February and March 2019, a replication of a survey carried out in Brazil [24] was conducted with practitioners engaged in Uruguayan software organizations. The primary goal of this study was the same as the previous study and refers to gaining knowledge about the perception of TD metaphor and its management (i.e., the adopted strategies, activities, and technologies) in software organizations, using the engaged practitioners as proxies. However, such a replication added the characterization of startups organizations in Uruguay. The survey was disseminated among personal contacts, more than 500 software practitioners subscribed to the IS.uy[1] mailing list, social networks (Facebook, LinkedIn, and Twitter), as well as through Information Technology (IT) associations (CUTI[2] and Uruguay XXI[3]).

As in its original lab package, the survey is structured in 14 sections: three characterize the participant, the organization, and the software project; one regards the TD perception; one in TDM in general; 8 relate to the specific TDM activities; and one extra section that provides space for the participant to describe other activities that are executed in the organization. The questionnaire sections are presented in Table 2. Three hundred and ninety-six participants answered this survey, with 259 complete answers. Considering the complete answers, 33 (13%) informed to work in startups. Therefore, all the results and discussions presented in the following sections are based on those 33 complete responses.

Aiming to identify the participants who work in a startup context, the questionnaire provided the following description of the startup concept: "*A temporary people organization (company or team) that generally presents most of the following characteristics: (i) activity in highly innovative segments of the market;(ii) conditions of extreme uncertainty regarding the success of the product and/or the business model; (iii) focus on a single main product; (iv) organizational structure to scale quickly in case of product success or dissolve otherwise*." This definition allowed unifying

[1]Uruguayan Software Engineering Community.

[2]http://www.cuti.org.uy/portada

[3]https://www.uruguayxxi.gub.uy/es/

Table 2 Questionnaire sections

Section	Topic	Description
1	Participant characterization	Obtain personal information regarding the participant, such as professional experience and academic degrees
2	Organization characterization	Gather information about the organization the participant works for or has worked before
3	Project characterization	Obtain information about the project that will be considered by the participant in the survey
4	TD perception	Collect information on the participant's knowledge regarding TD, including what can be considered TD. Also, determine if the organization or the project he works at has strategies for TDM
5	TDM (general)	Ask the participant which TDM activities are adopted in the working project. Obtain information about the responsibilities and importance associated with each activity from the participant's point of view
6–13	TDM (activities)	Gather information on several aspects regarding each of the TDM activities
14	TDM (other)	Provide space for the participant to describe other activities that are executed in the organization

the startup's concept among the participants, avoiding different interpretations. It should be noticed that this definition also admits considering a startup as a team inside a large company with a large number of employees.

Figure 1 summarizes the survey responses, considering the perspectives adopted in the analyses.

The respondents have an average work experience of 7 years in software projects. Regarding the academic degree, 20 respondents (61%) hold at least an undergraduate degree, and 6 of them reported holding a specialization, master or doctorate. The

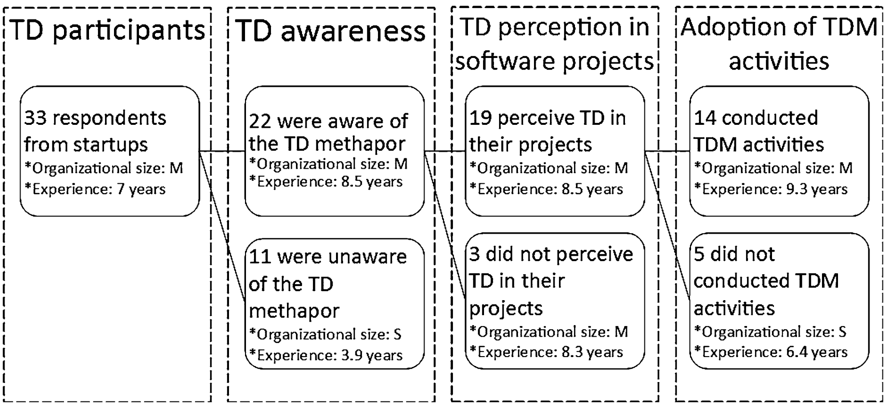

Fig. 1 Survey responses (only regarding startup organizations)

rest of them have at least an uncompleted undergraduate degree. The respondent's organizations present some of the typical characteristics of startups organizations [7], such as 24 respondents (73%) reported their organization has fewer than 50 employees and 22 participants reported that the development process adopted is agile or incremental. Among investigating Uruguayan software organizations, the majority of them (26) come from the IT sector, with a few respondents from Finance (2), Telecommunications (1), Biological/Pharmaceutical (1), Education (1), Consulting (1), and Commerce (1) sectors.

Regarding the perception of TD, 11 respondents (33%) claimed to be not aware of the TD metaphor, while the remaining 22 (67%) declare to understand the concept and relate it to the TD items that best match the TD definition (Table 3). Although most of the respondents reported being aware of the TD concept, it is not possible to observe any consensus about which issues are related to TD concept (no issue achieves 100% of the answers). However, 73% of the respondents who declared to be aware of TD agreed that the concept of TD could be associated with internal software quality problems. Even so, 41% of the respondents associate TD with external quality problems, which is worrisome since it contradicts the common understanding of the concept currently adopted in the academy.

From the 22 respondents claiming to be aware of TD meaning, 19 of them (86%) reported that they perceive the existence of TD in their projects, while 3 informed not perceiving it. From 19 respondents that informed to perceive the occurrence of TD in their projects, 14 (74%) stated that their organizations or the project managers carry out TDM activities in their projects. Table 4 shows the distribution of the counting of the votes (answers) on the different types of TDM activities.

Considering the 14 respondents that informed to adopt TDM activities in their projects, 9 of them stated that they also carry out TD prevention activities in their projects. Nine reported to conduct TD identification and TD payment; seven reported to conduct TD communication and TD prioritization, TD documentation

Table 3 TD perception by the respondents from startups organizations

Issue	% of participants
Low internal quality aspects, such as maintainability and reusability	72.73
Architectural problems (like modularity violation)	68.18
"Shortcuts" taken during design	63.64
Presence of known defects that were not corrected	54.55
Trivial code that does not violate code rules	50
Poorly written code that violates code rules	50
Code smells	45.45
Low external quality aspects, such as usability and efficiency	40.91
Planned, but not performed, or unfinished, tasks (e.g., requirements specification, models, test plans, among others)	31.82
Required, but unimplemented, features	31.82
Lack of support processes to the project activities	18.18
Defects	13.64

Table 4 TDM activities in software startups in Uruguay—technologies and strategies

TDM activity	Technologies and strategies
TD identification (9)	Manual code inspection (8), dependency analysis (5), checklist (1), Code coverage (1), and Architecture Analysis (1)
TD documentation/representation (4)	TD backlog (3), specific artifacts for TD documentation (1), JIRA(2), Wiki (1), others—Trello (1)
TD communication (7)	Discussion forums (2), TD topic in project meetings (6), specific TD meetings (3)
TD measurement (2)	Manual measurement (1), JIRA(1)
TD prioritization (7)	Cost/benefit analysis (3), specific technology for decision-making (1), classification of issues (4)
TD repayment (9)	Refactoring (8), redesign (5), rewrite of code (7)
TD monitoring (2)	Manual monitoring (1), JIRA(2), Wiki (1)
TD prevention (9)	Guidelines (4), coding standards (7), code revisions (9), retrospective meetings (7), Definition of done (3)

is conducted in projects according to four respondents, while just two respondents indicated the conduction of TD measurement and TD monitoring in the projects.

An interesting result refers to the experience (in years) of the participant (see Fig. 1). It increases concerning the awareness of the concept of TD; the perception of TD in projects; the management of TD by the organization; and the management of TD when the project manager is conducting extra activities. Another interesting observation is that the central tendency (mode) of the organizational size is small (S) when the participant declares not to be aware of the TD concept or declares that the organization does not carry out TDM activities.

Regarding the Responsibilities and Importance of Each TDM Activity There is no observed consensus among the participants on which TDM activity is most important as well as on which roles should play each TDM activity. Considering the relevant TDM activities to 14 respondents, 64% reported that TD identification and TD prevention are the essential activities to their projects, followed by TD communication and TD measurement. The TD documentation and TD monitoring were reported as least significant.

TD Identification From nine respondents that informed that TD identification is conducted in their projects, only two answered that there is a formal and mandatory strategy to conduct this activity; while seven participants answered that the activity is conducted informally. Four out of nine answers reported that TD identification was conducted continuously throughout the project, and five participants answered that they identify TD whenever a problem comes up. One participant informed that the TD was classified as Test Debt, Code Debt, Defect Debt, or Test Automation Debt in the project, other participants claimed to use the artifact where TD was detected, one answered informed to use TD dimension, and one reported to use other classification.

TD Documentation One out of four participants answered that they had a mandatory strategy to conduct the TD documentation, while three participants answered that they had a formal and mandatory strategy. Two out of four participants that answered the TD documentation section claimed that TD was documented using an overall backlog without specific details, and the other two informed that using a specific backlog for TD documentation.

TD Communication and Measurement From the seven participants that answered the TD communication section, three of them affirmed that the issues related to TD were discussed during meetings in their projects, but with the participation of few of the necessary stakeholders. Only two reported that issues related to TD were discussed in project meetings with all the stakeholders. Two participants answered that issues related to TD were only discussed informally. Only two participants answered the TD measurement section and say that the activity was conducted informally, based on simple information, like story points.

TD Prioritization Three out of seven participants answered that the TD items were prioritized according to "guesses" or simplified estimative based on previous experiences, two informed that the prioritization was based on a specific technology, one says that it was based on historical data, while the other participant used previous experiences and the priorities for the client. The distribution of number of answers (#) on what is the main criteria to prioritize the TD is: (7) TD items that most impact the project; (6) TD items that could cause most impact the client or has the highest level of severity; (5) TD items that are in used parts of the system; (4) TD items subject to immediate development or maintenance; (2) TD items that demand less effort to be paid or that have poor cost/benefit relation; and (1) based on the complexity of fixing the item.

TD Repayment, Monitoring, and Prevention From the nine participants that answered the TD repayment section, seven of them answered that the TD repayment is planned according to the current project necessities, while one of the participants answered that the TD repayment is planned continuously, with specific periods during the development process destined to this activity. Only two participants answered the TD monitoring section and informed that the activity was conducted only based on the number of technical debt items, or it is occasionally conducted. Finally, six of nine respondents answering the TD prevention section mentioned that it is an activity conducted informally by each project member. The remaining three informed that they had formal practices to conduct the activity in which two respondents say that these are mandatory practices.

Technologies and Strategies for TDM Table 4 presents a list of practices, techniques, and tools used in each TDM activity as reported by Uruguayan practitioners. The numbers in parentheses represent the number of participants answering that specific section (column "TDM activity") and the number of participants that affirmed using that technology or strategy (column "Technologies and strategies"). We can observe that different technologies support TDM, and there is no consensus about which one to use.

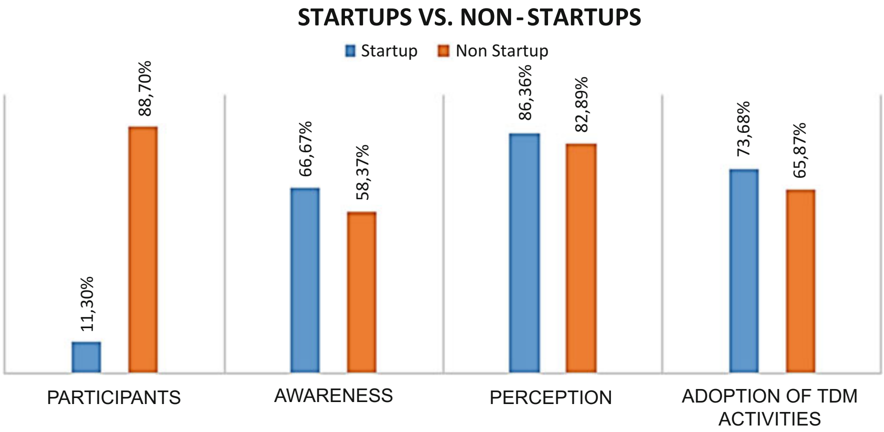

Fig. 2 Comparative graph startups versus non-startups

From the 259 complete answers obtained in the survey, 226 of them refer to the respondents whose organizations do not classify as a startup ("non-startup"). Figure 2 shows a comparison between startups and non-startups regarding the dimensions of the awareness of TD, the perception of TD in their projects, and the adoption of TDM activities. The results show that startup's participants have a little higher level of understanding and perception of TD, and they present a higher level of adoption of TDM activities.

4 Discussions

The study's results show some alignments in the views on TD between the study's participants and academia. Most of the participants agreed with the definitions from the technical literature, associating the TD concept to issues related to internal software quality issues. However, it was observed that there is no common perspective among the startup's practitioners about which issues relate to TD. A clear and shared understanding of the TD concept represents the first step toward the perception and management of TD by practitioners in their software projects.

As explained in Sect. 2, startups' characteristics may bring some risks to internal software quality, since one of their primary goals is to experiment with a new business model through the ideas developed for the product and give adequate feedback about the new features. However, it was possible to observe that the surveyed practitioners take into consideration the prevention and management of TD. Regarding TDM, most of them (42%) informed to perform TDM activities in their projects. The most reported activities regard TD prevention, TD payment, and TD identification. These results indicate that Uruguayan startups invest effort and time (which may be valuable for their context) to prevent and keep under control the accumulation of TD. This finding is in line with discussions reported in the technical literature which mention that neglect the TDM can bring negative consequences, in which it can become the leading cause of a startup failure [5, 7, 9].

Regarding the comparison between startup organizations and non-startup organizations, it could be observed that a small difference between the levels of TD understanding, TD perception, and adoption of TDM activities exists.

Regarding TDM activities, a set of strategies and technologies were identified as being used by Uruguayan startups' practitioners to support TDM activities in their projects. It is possible to observe that the same strategy or software technology was informed to support more than one TDM activity, such as JIRA and Trello (see Table 4). However, the effectiveness and efficiency of such strategies and technologies in managing TD must be the object of further investigation.

Besides that, it was not possible to observe the factors that lead the surveyed projects to manage TD. However, they likely are some of the ones identified in [5, 6], such as the experience of developers, employee growth, software knowledge of founders, uncertainty, lack of development process, the autonomy of developers, and the increase in the number of users.

Also, it was not possible to observe which factors influence TDM, but it was possible to observe that those participants who reported performing TDM activities belong to more prominent organizations (in the number of employees) than those who reported not performing TDM activities. This finding can be related to the "Employee growth" organizational factor, which may influence the TD prevention. Also, it is possible to observe a similar finding in the study reported in [6], stating the need to adopt some software knowledge registration practices when they needed to incorporate new members to the team, in which it may minimize the TD risks.

Another interesting finding is that the participants' experience would seem to relate to the TD perception and its management. The most experienced participants were those who declared to be aware of the concept of TD, perceived problems related to it in their projects, and their organizations carried out activities to manage it. It is related to another finding "*More experienced (senior) software developers are more aware of and have accumulated more experience about the effect of introducing TD, compared to junior developers*" [5].

The survey results indicate that TD is perceived and managed in the Uruguayan startup's context. However, it was not possible to identify in which startup phases the management of TD becomes a concern. A possible conjecture is that organizational factors influence the management of TD in a startup. Also, it may differ from the practices adopted in mature software organizations since the level of TD can be considered an investment in the initial startup's stages. Thus, the useful matching between the startup phases and TDM activities is essential to support the software startups to adopt strategies to balance the benefits and the challenges of TD in their projects over time.

The survey presents some threats to validity [35], in which the main concern is the generalization of results. On one hand, the targeted sampling is non-probabilistic; it is not possible to determine a priori the population size and the expected total number of participants. Uruguay is not known as a robust startup ecosystem, like Silicon Valley, for example. Therefore, it is hard to generalize the results. The internal threat is mainly associated with the participants that might have misunderstood some terms and concepts of the questionnaire based on different experiences. Besides, there is also a construct threat of a biased survey, from the researchers' perspectives and the collected information from the technical literature such as the TDM activities organized in [14]. Aiming to reduce the level of these menaces, we conducted revision cycles during the survey development with three researchers and executing pilot trials.

5 Concluding Remarks

In the last years, TD has been an exciting topic in the software engineering community by both practitioners and researchers. A particular and interesting context in which TD can emerge refers to software startups. This chapter presented background about the characterization of TD in software projects, the startup contextual characteristics that can lead to TD incurrence, and the results of an empirical study about the perception and management of TD under the perspectives of startups' practitioners.

The results indicate no unanimity concerning on how startups' practitioners perceive TD, at least at the context of startups in Uruguay. However, despite the uncertainty about their products and the speed in which they must validate the business model in the market, the surveyed startups seem to invest time and effort in TDM activities being the prevention of TD considered the most relevant TDM activity. Most of the participants that declare to understand and perceive the TD in their projects are distinguished by being those with the highest level of experience, and the participants who declare performing TDM activities belong to prominent organizations. A set of strategies and technologies used by Uruguayan startups were identified. However, further research is needed to investigate how effective and efficient they are in different software engineering communities. As it has been observed, it is necessary more validated TDM proposals on which strategies best fit the startup's context, helping them to meet their objectives and not fail in the attempt, because of unmanageable TD.

Therefore, the results and discussions presented in this chapter intended to provide contributions to both industry and academia. Regarding the contributions to the industry, the results provide initial observations regarding how software startups in Uruguay (represented by their practitioners) perceive and manage TD in their projects. Besides that, sharing some concerns that startups could take into account to support the management of TD in their projects, such as the stages of their software products, the experiences of their employers, employee growth, the increase in the number of users and their development process. The main contributions to the researchers are related to the observed research gaps that need further investigation, for instance, strategies and guidance to support the startups to perceive and manage TD by considering the mentioned concerns.

Survey findings

- ✓ It was not observed a shared perspective among startup practitioners about which issues relate to TD.
- ✓ Most of the survey's participants associated the TD concept with issues related to internal software quality issues.
- ✓ The number of employees (organizational size) and the participant's experience would seem to influence the perception and management of TD in a software startup context.
- ✓ Software startups need to tailor strategies and guidance to help them to better perceive and manage TD, by considering the stages of their software products, the experiences of their employers, employee growth, the user size grows and their development process.

References

1. Tom, E., Aurum, A., Vidgen, R.: An exploration of technical debt. J. Syst. Softw. **86**(6), 1498–1516 (2013)
2. Avgeriou, P., Kruchten, P., Ozkaya, I., Seaman, C.: Managing technical debt in software engineering (Dagstuhl Seminar 16162). Dagstuhl Reports 16162, Schloss Dagstuhl—Leibniz-Zentrum für Informatik, Germany (2016)
3. Gralha, C., Damian, D., Wasserman, A. I., Goulão, M., Araújo, J.: The evolution of requirements practices in software startups. In: Proceeding of the 40th ICSE, Gothenburg, Sweden, vol. ICSE '18, pp. 823–833 (2018)
4. Yli-Huumo, J., Rissanen, T., Maglyas, A., Smolander, K., Sainio, L.-M.: The relationship between business model experimentation and technical debt. In: Software Business 210. Springer, Cham, pp. 17–29 (2015)
5. Besker, T., Martini, A., Lokuge, R., Blincoe, K., Bosch, J.: Embracing technical debt, from a startup company perspective. In: Proceedings of the IEEE 2018 ICSME, Madrid, pp. 415–425 (2018)
6. Nascimento, L. M., Travassos, G.: Software knowledge registration practices at software innovation startups: results of an exploratory study. In: Proceedings of the 31st SBES, Fortaleza, CE, pp. 234–243 (2017)
7. Chicote, M.: Startups and technical debt: managing technical debt with visual thinking. In: Proceedings of the 1st ICSE-SEIP, Buenos Aires, Argentina, pp. 10–11 (2017)
8. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: A systematic mapping study. Inform. Softw. Tech. **56**(10), 1200–1218 (2014)
9. Klotins, E., Unterkalmsteiner, M., Chatzipetrou, P., Gorschek, T., Prikladnicki, R., Tripathi, N., Pompermaier, L.: Exploration of technical debt in start-ups. In: Proceedings of the 40th CSE-SEIP, Gothenburg, pp. 75–84 (2018)
10. Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: the greenfield startup model. IEEE Trans. Softw. Eng. **42**(6), 585–604 (2016)
11. Cunningham, W.: The WyCash portfolio management system. In: Addendum to the proc. on object-oriented programming systems, languages, and applications, Vancouver, BC, vol. 4(2), pp. 29–30 (1992)
12. Ampatzoglou, A., Ampatzoglou, A., Chatzigeorgiou, A., Avgeriou, P.: The financial aspect of managing technical debt: A systematic literature review. Inform. Softw. Tech. **64**, 52–73 (2015)

13. Becker, C., Chitchyan, R., Betz, S., McCord, C.: Trade-off decisions across time in technical debt management: a systematic literature review. In: Proceedings of the 2018 International Conference on Technical Debt (ICSE), Gothenburg, Sweden, pp. 85–94 (2018)
14. Li, Z., Avgeriou, P., Liang, P.: A systematic mapping study on technical debt and its management. J. Syst. Softw. **101**, 193–220 (2015)
15. Alves, N., Mendes, T., Mendonça, M., Spínola, R., Shull, F., Seaman, C.: Identification and management of technical debt: A systematic mapping study. Inform. Softw. Tech. **70**, 100–121 (2016)
16. Alves, N., Mendonça Neto, M., Spínola, R.: A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners. Inform. Softw. Tech. **102**, 117–145 (2018)
17. Seaman, C., Guo, Y.: Measuring and monitoring technical debt. In: Advances in Computers, vol 82, pp 25–46. Elsevier (2011)
18. Klinger, T., Tarr, P., Wagstrom, P., Williams, C.: An enterprise perspective on technical debt. In: Proceedings of the 2nd international workshop on MTD, Waikiki, Honolulu, HI, USA, pp 35–38 (2011)
19. Spínola, R., Vetrò, A., Zazworka, N., Seaman, C., Shull, F.: Investigating technical debt folklore: shedding some light on technical debt opinion. In: Proceedings of the 4th international workshop on MTD, San Francisco, CA, USA, pp. 1–7 (2013)
20. Lim, E., Taksande, N., Seaman, C.: A balancing act: what software practitioners have to say about technical debt. IEEE Softw. **29**(6), 22–27 (2012)
21. Ernst, N., Bellomo, S., Ozkaya, I., Nord, R., Gorton, I.: Measure it? manage it? ignore it? software practitioners and technical debt. In: Proceedings of the 10th Joint Meeting on FSE, Bergamo, Italy, pp. 50–60 (2015)
22. Holvitie, J., Leppänen, V., Hyrynsalmi, S.: Technical debt and the effect of agile software development practices on it—an industry practitioner survey. In: Proceedings of the 6th International Workshop on MTD, Victoria, BC, Canada, pp. 35–42 (2014)
23. Martini, A., Besker, T., Bosch, J.: The introduction of technical debt tracking in large companies. In: Proceedings of the 23rd APSEC, Hamilton, New Zealand, pp. 161–168 (2016)
24. Silva, V., Jeronimo Jr., H., Travassos, G.: A Taste of the software industry perception of technical debt and its management in Brazil. In: Proceedings of the XXI CIbSE, Bogatá, pp. 1–10 (2018)
25. Sutton, S.: The role of process in software start-up. IEEE Softw. **17**(4), 33–39 (2000)
26. Dahlstedt, Å., Karlsson, L., Persson, A., NattochDag, J., Regnell, B.: Market-driven requirements engineering processes for software products—a report on current practices. In: International Proceedings of the Workshop on COTS and Product Software: Why Requirements Are So Important (RECOTS) (2003)
27. Hart, M.A.: The lean startup: how today's entrepreneurs use continuous innovation to create radically successful businesses, Eric Ries. J. Product Innov. **29**(3), 508–509 (2011)
28. Steve, B.: Why the lean start-up changes everything. Harv. Bus. Rev. **91**(5), 63–72 (2013)
29. Boer, H., Gertsen, F.: From continuous improvement to continuous innovation: a (retro)(per) spective. Int. J. Technol. Manag. **26**(8), 805–827 (2003)
30. Bosch, J.: Building products as innovation experiment systems. In: International Conference of Software Business, Berlin, pp. 27–39 (2012)
31. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown, New York (2011)
32. Crowne, M.: Why software product startups fail and what to do about it. Evolution of software product development in startup companies. In: IEEE International Engineering Management Conference, Cambridge, UK, vol. 1, pp. 338–343 (2002)
33. Berg, V., Birkeland, J., Nguyen-Duc, A., Pappas, I.O., Jaccheri, L.: Software startup engineering: A systematic mapping study. J. Syst. Softw. **144**, 255–274 (2018)
34. Ko, A.: A three-year participant observation of software startup software evolution. In: Proceedings of 39th ICSE-SEIP, Buenos Aires, Argentina, pp. 3–12 (2017)
35. Linåker, J., Sulaman, S., Maiani de Mello, R., Höst, M.: Guidelines for conducting surveys in software engineering. Tech Report TR 5366801, Lund University (2015)

Part II
Startup Engineering Methodology

An Analytical Framework for Planning Minimum Viable Products

Anh Nguyen-Duc

Abstract For early-stage high-tech startups, Minimum Viable Products are the most important artifacts for both business development and product development. In an entrepreneurial journey with build–measure–learn loops, startups need to be certain about what they learn to be closer to a product–market fit. Grounded from insights of 40 active digital startups, we proposed the 6W3H framework that captures a comprehensive set of context factors for developing an MVP. The framework represents an effectual MVP development with the relationships among the existing competence (Who question), business ideas (Why question) and current customers (For Whom questions), MVP's features (What to build question), Startup metrics (What to measure question), and the development processes and practices (How questions). We demonstrate how 6W3H framework can be used for visualizing startup development, supporting decision-making, and mitigating product risks. The benefits of using the framework are highlighted when MVPs associating with significant uncertainty and fast-changing requirements and team resources.

Keywords 6W3H framework · Minimum viable product · MVP context · MVP development · Multiple case study

1 Introduction

The term Minimum Viable Product (MVP) was defined by Frank Robinson [1] in 2001 and then popularized from Lean Startup movement by Eric Ries from 2009 [2] and Blank [3] from 2010. According to Lean Startup [2, 3], every startup should

A. Nguyen-Duc (✉)
Department of Business and IT, University of South-Eastern, Bø i Telemark, Norway
e-mail: angu@usn.no

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_5

start with building an MVP, and use it to validate their hypotheses about customer needs. It plays an important role not only for a startup team, but also for the startup's external stakeholders, such as potential users, investors, and mentors [4]. Together with "startup," MVP is one of the most overused and misunderstood terms among practitioners. Google search for the term "Minimum Viable Product" results in 1.4 million articles. Most of the entities in the first page discuss different definitions and interpretations. Among researchers, a literature review revealed that at least 22 different definitions and features of MVPs are used in Software Engineering research. Eric Ries states in his Lean startup book: "One of the most important lean startup techniques is called the MVP. Its power is matched only by the amount of confusion that it causes, because it is actually quite hard to do. It certainly took me many years to make sense of it." Steve Blank said about MVP: "This minimum feature set causes lots of confusion. Founders act like the 'minimum' part is the goal. Or worse, that every potential customer should want it" [3].

Despite the escalating complexity of debates around MVPs, entrepreneurs need to utilize them effectively in their startups. Until the time writing this chapter, we found no convincing and concrete guidelines for conducting MVPs in their development context. Many articles [5–7] base on expert opinions without explicit arguments for credibility or grounded process on empirical evidence. Many others (e.g., [8]) base on one or a few case studies that are questionable about their generalization. For some scientific articles [4, 9–16], the findings might have a limited implication for startups. It is desirable to come up with a simple guideline that can be used by everyone in a startup to improve the effectiveness of the MVP development journey.

This chapter aims at characterizing software MVPs and how to effectively plan for developing them in a lightweight manner. We propose a framework, namely 6W3H that describes contextual factors that influence the creation of an MVP. The framework's elements are grounded from 40 cases of European startups. We demonstrate for the use of the framework by three use cases: (1) visualizing evolution paths of startups, (2) a decision-making support mechanism, and (3) a risk mitigation framework. The chapter is organized as follows: Section 1 introduces about MVPs in research and practice; Sect. 2 describes the promised benefits of MVPs in startups; Sect. 3 presents a list of MVPs according to Eric Ries' categories; Sect. 4 proposes the 6W3H model, the model taxonomies, and its applications; Sect. 5 presents the demographics of our cases; and Sect. 6 concludes the chapter.

2 Benefits of MVPs

We start from Eric Ries' definition, as it is among the most popular concept: "a version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort" [2]. We classified

Table 1 Benefits of MVPs in early-stage startups

Benefits	Descriptions	Features
Resources optimization	MVPs save the effort/money of building fully functional products, hiring full-stack technology competence to validate a business idea [2, 5]	Least efforts
Early customer acquisition	Though MVPs lack some top-notch features and advanced functionality, it does provide value to users and hence acquires early adopters. Nevertheless, it does not prevent you from starting to work with key startup metrics and make up a customer base [2, 5]	Maximum amount of learning
Value proposition focus	The MVP lets you understand different problems your future customers need to solve. You can take advantage of using the value proposition canvas to get a graphical expression of customers' needs vs. product's offers [2]	Maximum amount of learning
Innovation facilitating	The continuous creation of certain types of MVPs helps to visualize design ideas, offers a chance to refine both architecture and design of a product [5]	A version of a new product
Reusability	Regarding evolutionary or throwaway MVPs, most of them can be reused in the next MVPs in one way or another [4, 17]	Least efforts
Communication artifact[a]	MVPs could be used during meetings with external stakeholders, acting as a bridge between business minds and technical minds, emphasized call for investment or involvement [4, 17]	Communication
Documentation[a]	MVPs imply the lesson learned after each build–measure–learn loop, hence storing knowledge and growth hacking mechanisms [4, 17]	Communication
Value co-creation[a]	Cooperation and hand-in-hand work with potential users in crafting the final product necessary [4, 17]	Communication
Call for funding[a]	Possibility to attract investors early and the ability to apply for crowdfunding [4, 17]	Communication

[a]New benefit beyond Eric Ries' definition

the benefits of an MVP based on the dimensions "a version of a new product," "maximum amount of learning," and "least efforts." Research about MVPs in software development reveals other benefits of MVPs in communicating, documenting, value co-creation, and call for funding [4, 17], as shown in Table 1. Different MVPs might be developed to bring different benefits. The managerial, financial, and business dimensions of these MVPs make them different from technical prototypes.

✓ **Minimum viable product (MVP)** is a proxy of the final product that requires the least effort to develop but obtain maximum learning. MVP is useful for project planning, product development, fundraising, and communication.

3 Different Types of MVPs

Eric Ries initiates the classification of MVP types [2], which are discussed among the community of practitioners, including:

- Explainer video: a short animation that explains what your product does and why users should buy it. The video is often simple, lasts for 30 s to a few minutes. The video is particularly useful to check out unique ideas that may not have been seen before. The most famous example of Explainer video is Dropbox.[1]
- Landing page: a web page where visitors "land" after clicking a link from an e-mail or another type of a campaign. A landing page is used to quickly communicate the startup proposals, to diffuse objections, and to call the visitors to action. With the help of a landing page, you can get early followers and collect a potential user base, with a limited budget. A landing page needs to be structured with proper headlines, value propositions, and call-to-action to test the business idea.
- Wizard of Oz: a user interface that looks like a real working product, but the actual business process is manually carried on. The purpose of this MVP is to demonstrate the complete job done by the product. For hardware products, this would be equal to "looks like" kind of prototypes.
- Concierge MVP: a manual service that consists of exactly the same steps users would go through with the product. This is called a "concierge," since you need first to provide services manually. For instance, instead of displaying personalized news for readers, you go through their preferences and reading history manually, and show the news you find relevant.
- Piecemeal MVP: similar to Wizards of Oz MVP, however, execution of the tasks is done by using existing tools. It collects the necessary components and pieces them together in a way that gives a new functionality and user experience.
- Mockup MVP: such as paper prototypes and wireframes, was representative of user interfaces without any functionality. Various tools for sophisticated simulations of user interfaces, screen flows, and human interactions are available, i.e., Balsamiq,[2] Visual Paradigm,[3] JustInMind,[4] and InVisionApp[5].
- Public project proposal: Kickstarter[6] and other crowdsourcing sites, i.e., GoFundMe,[7] and Indiegogo,[8] allow for users to prepurchase the product and

[1] https://www.youtube.com/watch?v=w4eTR7tci6A
[2] https://balsamiq.com/
[3] https://www.visual-paradigm.com/
[4] https://www.justinmind.com/
[5] https://www.invisionapp.com
[6] https://www.kickstarter.com/
[7] https://www.gofundme.com/
[8] https://www.indiegogo.com/

provide a great way to raise money for initial orders. A business idea can be validated by feedbacks and fund contributed to the call in such crowdfunding sites.

- Single-feature MVP: a prototype that implements the most important function of the product. The feature is normally implemented with a certain level of quality and functionalities. The prototype is probably the most closer one to working products since it is the first working version of the product with only a core feature to attract early adopters.
- Rip off MVP: a successful product to get feedback, and then pivot in a different direction.

4 The 6W3H Framework of Building an MVP

4.1 The Motivation for the Framework

The Five W's and How method is widely used in journalism, research, and police investigation for context analysis. By emphasizing multiple dimensions of a context, the framework constitutes a formula for achieving a complete story on a subject. Thomas Aquinas had acknowledged Aristotle as the originator of the dimensions of circumstances [11]. He examined the concept of Aristotle's voluntary and involuntary action by investigating the question "Whether a circumstance is an accident of a human act." In his article, he mentioned, "For in acts we must take note of who did it, by what aids or instruments he did it (with), what he did, where he did it, why he did it, how and when he did it" [11]. Beyond journalism, characterizing contextual elements is commonly used in problem analysis, project management, and software engineering research [18]. For instance, Tore Dybå et al. argue for the shift of focus from a checklist-based approach to a context in favor of a more dynamic view of software engineering practice. Instead of viewing context as a set of discrete variables that statically surround parts of a practice or an artifact, we should treat context in its reflexive relationship with practices and artifacts [18]. The interpretive work a practice/artifact generates shapes context as much as the context shapes the practice/artifact [18]. Consequently, by considering an MVP as an artifact, and the MVP creation an engineering process, it is necessary to characterize the relationship between the MVP and its context. We argue that six W-questions and three H-questions are relevant for characterizing the context of MVP development.

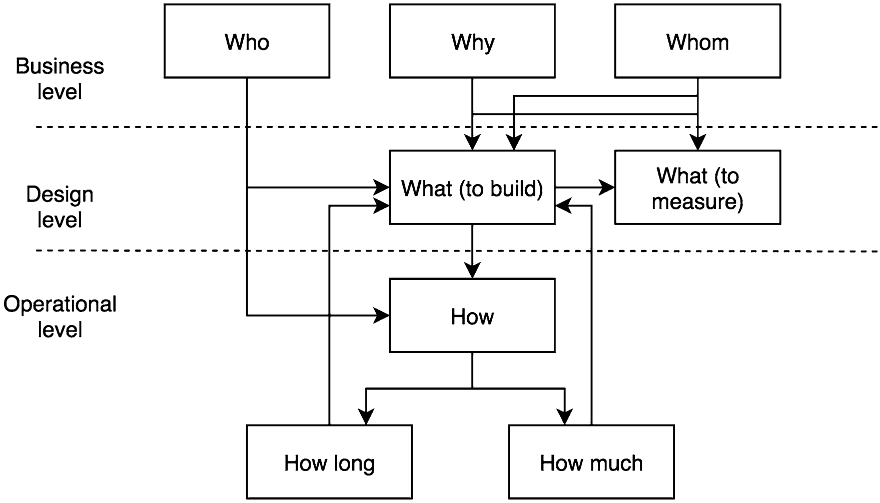

Fig. 1 6W3H frameworks of building MVPs

4.2 *The Descriptions of the 6W3H Framework*

In this section, we describe the mapping of 6W3H frameworks of startup context to Lean startup methodology [2]. We explain how each of the W element and H element links to the central concept and to each other, as shown in Fig. 1.

What The What question is split into two sub-questions, namely (1) what to build and (2) what to measure, to emphasize the preparation needed for both building and measuring. Our empirical cases (as details in Sect. 5) show that startups rarely measure. Instead, decisions are often made by entrepreneur's god feelings or experience, reacting to current conditions of competitors and markets. However, according to Lean Startup, data should be a major source for making decisions. The specification of the MVPs would be documented in various formats, business goals, customer emails, use cases, user stories, formal specifications, competitor's products, etc. Answering the What question is the journey of moving from unknown to known domains, with the later MVPs be closer to the customer/market needs. The plan for MVPs should come also with its measurement. Goal-question-metric [19] is a good way of deriving the list of metrics from the Why question.

Why Before investing effort and time on elaborating various MVPs, we would check first whether hypotheses behind the MVPs are established. There might always be some thoughts behind what to do with the MVPs and good feelings are usually the way to determine if the learning is achieved. A proper formulation of hypotheses comes with a plan to test them, hence, giving input to both What to build and What to measure. Effectual startups might not proceed with one long-term Why question in the beginning, but move forward by answering a series of short-term Why that the later one is the consequence of the previous one.

Who Many startups gather necessary competence before they start getting their hands dirty. Many others start right away with what currently in their hands. An example is a businessman/woman that is aware of an entrepreneurial opportunity. They used a limited budget to gather necessary competence that usually not the optimal one for their startups. Another example is a technical person, i.e., researchers, engineers, and professors that holds some advanced technology that would like to commercialize them. The business competence is often missing in such a case. It is not uncommon that hiring and product developing occur in parallel, leading to the later MVPs is done better than early ones due to the available competence. Besides, external competence, i.e., freelancers, outsourcing partners, is also relevant to MVP development. Hence, *Who* is in the team has a direct impact on What and How matters.

For whom At the time of developing the MVPs, startups might already involve some external stakeholders who influence the specifications of the MVPs. They can be customers who paid for a customer-bespoken development of products, lead users who are in the frontier of the product/market segments. They are input sources for What to build and measuring objects for What to measure.

When The temporal dimension represents the amount of accumulated learning a startup achieves. The expected learning should be the input for deciding what to build and to measure. Early-stage MVPs are often low-fidelity ones, such as wireframes, landing pages, and mockups. Later stages involve high-fidelity MVPs, such as single-feature MVPs or rip off MVPs.

How The question specifies the strategy, methods, processes, and techniques to realize the planned MVPs. The development of these MVPs can be done in-house or by external resources. Here we focus on the more complicated MVPs, such as functional MVPs involving software development, hardware development, and industrial design. From this perspective, the construction of an MVP could be framed as a product development project, with dimensions of scope, time, and cost. The rule of thumb is that only two of the three dimensions can be optimized. The balance among these dimensions is represented by the two-way links among questions What to build, How long to build, and How much to build.

Depending on whether the MVP or the process of MVP development is of interest, the What question or the How question would stay in the center of the framework. It is noted that the MVP and its development has a mutual impact on each other. As the focus of this chapter is the MVP as an artifact, we investigated other Ws and Hs as surrounding contextual elements. However, different varieties

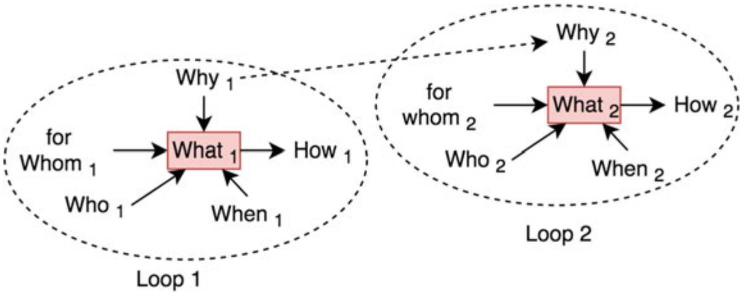

Fig. 2 The evolution of MVPs via 6W3H framework

of the framework can set How questions in the center too. Figure 1 visualizes the links among W-elements and H-elements we described above together. The Who, the Why, and the Whom is at the business level, usually not changeable for software developers. The What represents the analysis and design activities of software developers. The How shows the operationalization of the design ideas.

We argue that a build–measure–learn loop can be planned, visualized, and managed by this 6W3H framework. The hypothesis of business or product idea is established when answering the Why question. The plan for what to build and how to build is made with considering Who and For Whom questions. The measurement is also captured before building the MVPs. A change happening to one or more elements of the framework would imply the revisiting of other Ws and Hs for the optimal balance of the context and the developing MVP. The continuous awareness and analysis of the context elements would give a useful mean for visualizing and managing the evolution of a startup. As shown in Fig. 2, the first build–measure–learn loop might finish when the construction and learning from the MVP is done. The next loop would occur on top of learning and materials done in the first loop, addressing the new Why, and building the new MVP.

A brief of the 6W3H framework

- ✓ What should the MVP include?
- ✓ What should be measured with the MVP?
- ✓ Why should the MVP be built?
- ✓ Who will build the MVP?
- ✓ For whom the MVP will be built?
- ✓ When will the MVP be built?
- ✓ How will the MVP be built?
- ✓ How much does it cost to build the MVP?
- ✓ How long does it take to build the MVP?

4.3 Three Use Cases of 6W3H Framework

4.3.1 Use Case 1: Visualizing the Evolution Path of the Startup Journey

Startup A[9] develops a platform for modeling, communicating, and document management in a construction project. The startup originates from Norway and currently has more than 5000 customers in Scandinavian countries, Poland, Germany, and the USA. The company adopts a Business-to-Business model, delivering a digital platform for construction projects. Initiated by a serial entrepreneur in 2013, Startup A represents for a successful Norwegian startup with stable revenue and customer growth. It is special that the company starting with no in-house software developer. The progress of the first 12 months of operation can be characterized via three featured MVPs, as shown in Table 2. It is shown that A has established early a relationship with an outsourcing team. Analysis and design are done in-house via strong collaboration between the CEO and an early paying customer.

4.3.2 Use Case 2: Decision-Making Support

Startup B[10] develops a platform for hyper-local news. The startup originates from Norway and testing markets in Singapore and Vietnam. The company was founded by two members. The CEO has a journalism and marketing background and the CTO has a software development background. The startup also employed a Vietnamese outsourcing team to develop their MVPs. The first MVP was created in late 2015. The product launched at the end of 2015. In Startup B, startup methodology was emphasized since the early days. The CTO attempted to formally adopt Lean Startup and Agile in product development. In the first 3 months, the CEO stated the first hypothesis (as stated in Loop 1, Table 3) that people in the current local regions are interested in some types of news within a radius of 3 km around them. As shown in Table 3, three alternative approaches are available for testing the hypothesis a Concierge MVP by using a Facebook page, a simple one-feature MVP and an evolutionary functional MVP. The CEO also considered the second hypothesis (as stated in Loop 2, Table 3) that the target user groups would rather read news in a map view than read them in a list view. When coming to cost and time, Alternative 3 seems to be advantageous as it saves other cost and focus right away on the MVP implementation. Moreover, the team can start early with the development with a lot of reuse for future MVPs. The team did use Alternative 3, and unfortunately stuck in a problem that product is developed without market validation. The hypothesis in Loop 1 should be validated before any technical implementation with the outsourcing team. The postmortem analysis conducting

[9] http://viscenario.com/

[10] https://newsinitiative.withgoogle.com/dnifund/dni-projects/muml/

Table 2 Three featured MVP in the first 12 months of A's operation

	Loop 1	Loop 2	Loop 3
Why	To address the need of a specific customers To verify the needs for more than one customers	The feature is the most important pain point for the customer	The feature is suitable for other customers
Who	The CEO	The CEO, an Indian outsourcing team	The CEO, an Indian outsourcing team
Whom	A paying customer	A paying customer	A mass market
What to build	Mockup MVPs	One-feature functional MVP	Evolutionary functional MVP
What to measure	Qualitative feedbacks from management board	Qualitative feedbacks from management board	Qualitative feedbacks from several companies
How	In-house design, co-creation with customers	In-house design, outsourced development	In-house design, outsourced development

Table 3 Alternative approaches for developing MVPs in two consecutive learning loops

Whom	Mass market		
What to measure	– amount of posts per day – number of readers	– posts per day velocity – reader velocity	– posts per type – posts per location
	Alternative 1	*Alternative 2*	*Alternative 3*
Who	The CEO	The CEO, an in-house developer	The CEO, an outsourcing team
How	In-house setup	In-house development	Outsource-development
Loop 1			
Why	People are interested in hyper-local news around them		
What to build	Concierge MVP Community of readers	MVP containing a simple list view of posts Community of readers	MVP containing a list view of posts Community of readers
How long	Few hours for setting up a Facebook page 3 weeks to engaging the community	Three days to make the simple one-feature MVP 3 weeks to engaging the community	Two weeks to set up and to build the evolutionary MVP 3 weeks to engaging the community
How much	2000 NOK	4000 NOK	8000 NOK
Loop 2			
Why	People like to read news in a map view than in a list view		
What to build	MVP containing a list view and a map view of posts	MVP containing a list view and a map view of posts	Adding the map view to the current MVP
How long	Three weeks to set up and to build	Three weeks to set up and to build	One week to set up and to build
How much	16,000 NOK	16,000 NOK	8000 NOK

with both the CEO and the CTO of B revealed that they would have followed Alternative 1 before engaging too much on product development.

4.3.3 Use Case 3: Risk Mitigation Framework

We continue with Startup B and use the 6W3H framework to identify possible risks when validating business hypotheses. We argue that not only the hypotheses, but also the means of testing hypothesis need to be validated too. Criticizing the connections from What question to other Ws would reveal assumptions that need to be tested. Entrepreneurs should attempt to falsify these assumptions they survive until the next test.

These include:

- What–Why: is the intended MVP the right thing to test the given hypothesis?
- What–Why: is the selected metrics the right measurement for the given hypothesis?
- What–Whom: is the intended MVP suitable to all of the relevant stakeholders?
- What–Who: is the current competence the right one to build the MVP, in terms of cost and quality?
- What–When: is the intended MVP suitable for the current stage of the startup journey?
- What–How: is the chosen approach the right one to build the MVP, in terms of cost, quality, and future reuse?

In the same postmortem analysis mentioned in the previous section, we revealed many decisions without convincing supports, for instance:

What–Why: is the intended MVP the right thing to test the given hypothesis?

- The CEO needs to decide which type of MVP they should use. Prior to the consideration of who, how questions, the first concern is whether the selected MVP suitable to test the given hypothesis. For instance, if the selected social media does not show the location, it is not possible to test the map view feature with this concierge MVP.

What–Why: is the selected metrics the right measurement for the given hypothesis?

- There is a risk that the metric is not able to test the hypothesis. Are users posting in the MVP representative for the intended customer segment? Does the number of posts per day actually show the interest of people in hyper-local news?

What–Whom: is the intended MVP suitable to all of the relevant stakeholders?

- There are many cases that the inputs for MVPs coming from various stakeholders, including mentors, investors, big customers, and smaller customers.

It is a risk that the intended MVP does not fit to all stakeholders. Prioritization of features is needed to mitigate this risk.

What–Who: is the current competence the right one to build the MVP, in terms of cost and quality?

- Once the type of MVPs and features of the MVP is determined, there is a risk that current competence is not capable of making the MVPs. For instance, an MVP of an Internet of Things platform with the focus on optimizing communication latency would require a research duration, and hence imply a risk of completing the feature.

What–When: is the intended MVP suitable for the current stage of the startup journey?

- It is a question of whether introducing the outsourcing team in idealization and conceptualization is a reasonable decision. Should it be more business hypotheses validated with the core team before engaging in more technical implementation?

What–How: is the selected approach the right one to build the MVP, in terms of cost, quality, and future reuse?

- For building the same MVP, it is always challenging to select a development approach that balances the current quality, cost, and those in future releases.

4.4 The Taxonomy of Contextualizing the MVP Development

What Is a Taxonomy? By taxonomy, we mean "A system for naming and organizing things [. . .] into groups which share similar qualities" (Cambridge Dictionaries Online). Such a taxonomy can be used for a wide variety of purposes. Among others, the taxonomy can help entrepreneurs choosing a suitable approach for building their MVPs given a specific context. But first, the taxonomy presenting here is used to classify different context of MVP development. The selection of one or more elements in each Ws and Hs categories in Fig. 3 represents a situation that an MVP is developed, hence revealing an opportunity for optimizing. The taxonomy complements for the 6W3H framework.

Where Does the Taxonomy Come From? The framework and taxonomy are grounded from 40 active European startups, as described in detail in Sect. 5. From each startup, we identified a few key MVP and explored their contextual dimensions.

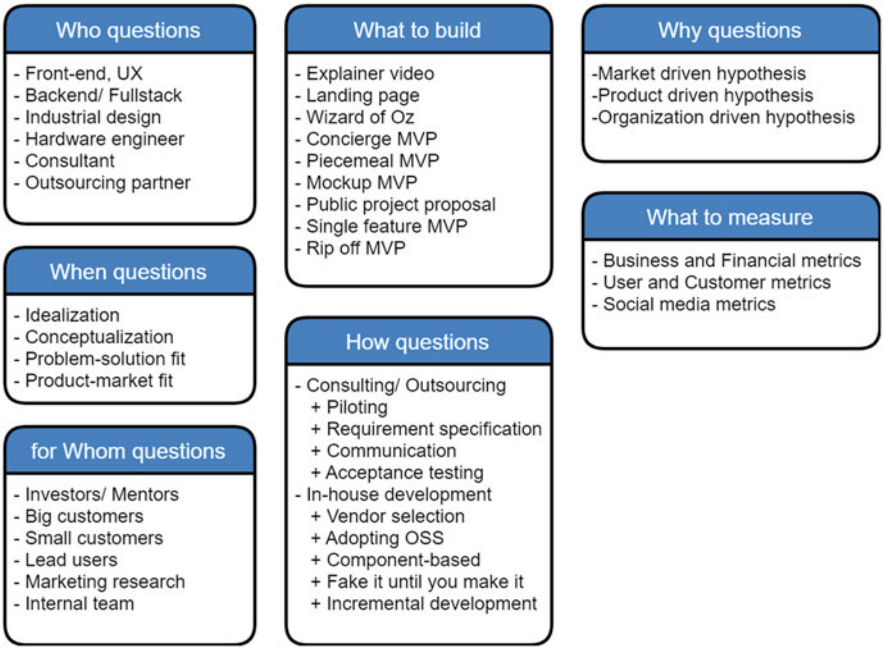

Fig. 3 Taxonomies used in 6W3H framework

5 Our Cases

There is often a difficulty in identifying a real startup case among other similar phenomena, such as freelancers, SMEs, or part-time startups. We defined five criteria for our case selection:

1. A startup that has at least two full-time members, so their prototyping practices are not individual activities
2. A startup that operates for at least 6 months, so their experience can be relevant
3. A startup that has at least a first running prototype, so the prototyping practices is a relevant topic
4. A startup that has at least an initial customer set, i.e., first customer payments or a group of users, so that certain milestones in startups process is made
5. A startup that has software as the main part of the business core value

The process of identifying and collecting data was done in 18 months, from March 2015 to June 2018. From a contact list of 300+ startup companies, we selected 40 startups that are suitable to the research purpose and willing to participate in the research project (including Cases A and B above). The contacted startups come from Norway, Sweden, Finland, Italy, Germany, Spain, the Netherlands, Singapore, India, China, Pakistan, China, and Vietnam. Semi-structured interviews were one of the main data collection methods. The unit of analysis is a startup company. We intended to have multiple interviews in each startup, to achieve triangulation in data [9]. However, most of the startups only afford a single interview. It is noted that some parts of the data set have been explored in previous research [4, 17, 20].

6 Conclusions

For early-stage high-tech startups, MVPs are the most important artifacts for both business development and product development. In an entrepreneurial journey with build–measure–learn loops, startups need to be certain about what they learn to be closer to a product–market fit. For taking the most out of an MVP, the analysis of its context is needed. In this chapter, we proposed 6W3H framework that characterizes nine important contextual dimensions of an MVP. The taxonomy using in the framework was systematically gathered from 40 active high-tech startups. This is a result of a long-term project with 2 years of data collection. The framework represents an effectual view of the MVP development by connecting the existing competence (Who questions), business ideas (Why question), and current customers (For Whom questions) to the desired MVP. Moreover, the framework emphasizes the measurement of learning (What to measure question), which is often neglected in early-stage startups. Last but not least, the framework presents a reflexive relationship between the MVP (What questions) and the MVP development (How questions).

We have demonstrated three use cases of 6W3H framework, but the potential use is unlimited. The most important takeaway lesson for entrepreneurs is that the construction, feedback collection, and learning for an MVP needs to be planned with clarification of possible assumptions, and prepared for changing of the MVP's contextual elements. Understanding this, entrepreneurs would be better in knowing and communicating their startup development strategies, making thoughtful decisions, reducing avoidable risks, and being closer to successes.

Key findings

- ✓ MVPs should always be planned with clarification of contextual assumptions.
- ✓ The mutual relationships among the element of 6W3H and their evolutions are complex over time.

References

1. Robinson, F.: A Proven Methodology to Maximize Return on Risk (2001). http://www.syncdev.com/minimum-viable-product. Accessed Mar 2019
2. Ries, E.: What is the Minimum Viable Product (2009). http://venturehacks.com/articles/minimum-viable-product. Accessed Mar 2019

3. Blank, S.: Perfection by Subtraction – The Minimum Feature Set. (2010). http://steveblank.com/2010/03/04/perfection-bysubtraction-the-minimum-feature-set/. 2010. Accessed Mar 2019
4. Duc, A.N., Abrahamsson, P.: Minimum viable product or multiple facet product? The role of MVP in software startups. In: Sharp, H., Hall, T. (eds.) Agile Processes, in Software Engineering, and Extreme Programming, pp. 118–130. Springer International, Cham (2016)
5. What Is A Minimum Viable Product, and Why Do Companies Need Them. (2018). https://www.forbes.com/sites/quora/2018/02/27/what-is-a-minimum-viable-product-and-why-do-companies-need-them/#2720ace7382c. Accessed Mar 2019
6. Definition of "Minimum Viable Product". (2019). https://economictimes.indiatimes.com/definition/minimum-viable-product. Accessed Mar 2019
7. What is a Minimum Viable Product and How to Build an MVP for Your Startup. (2018). https://medium.com/@sprocompany/what-is-a-minimum-viable-product-and-how-to-build-an-mvp-for-your-startup-9a02c0d4a56a. Accessed Mar 2019
8. Guthrie, G.: How Minimum Viable Products Can Kick-Start Your Startup. (2018). https://backlog.com/blog/how-minimum-viable-products-can-kick-start-your-startup/. Accessed Mar 2019
9. Boyatzis, R.E.: Transforming Qualitative Information: Thematic Analysis and Code Development. Sage, Thousand Oaks (1998)
10. Lenarduzzi, V., Taibi, D.: MVP explained: a systematic mapping study on the definitions of minimal viable product. In: 2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 112–119. (2016). https://doi.org/10.1109/SEAA.2016.56
11. Münch, J., Fagerholm, F., Johnson, P., Pirttilahti, J., Torkkel, J., Jäarvinen, J.: Creating minimum viable products in industry-academia collaborations. In: Fitzgerald, B., Conboy, K., Power, K., Valerdi, R., Morgan, L., Stol, K.-J. (eds.) Lean Enterprise Software and Systems, pp. 137–151. Springer, Berlin (2013)
12. Anderson, E., Lim, S.Y., Joglekar, N.: Are More Frequent Releases Always Better? Dynamics of Pivoting, Scaling, and the Minimum Viable Product. HICSS (2017)
13. Fagerholm, F., Sanchez Guinea, A., Mäenpää, H., Münch, J.: The RIGHT model for continuous experimentation. J. Syst. Softw. **123**, 292–305 (2017). https://doi.org/10.1016/j.jss.2016.03.034
14. Nguyen-Duc, A., Khalid, K., Shahid Bajwa, S., Lønnestad, T.: Minimum viable products for internet of things applications: common pitfalls and practices. Future Internet. **11**(2), 50 (2019). https://doi.org/10.3390/fi11020050
15. Hyrynsalmi, S., Klotins, E., Unterkalmsteiner, M., Gorschek, T., Tripathi, N., Pompermaier, L.B., Prikladnicki, R.: What is a minimum viable (video) game? In: Al-Sharhan, S.A., Simintiras, A.C., Dwivedi, Y.K., Janssen, M., Mäntymäki, M., Tahat, L., Rana, N.P., et al. (eds.) Challenges and Opportunities in the Digital Era, pp. 217–231. Springer International, Cham (2018)
16. Agile Prototyping for technical systems–Towards an adaption of the Minimum Viable Product principle. Accessed Mar 2019
17. Nguyen-Duc, A., Wang, X., Abrahamsson, P.: What influences the speed of prototyping? An empirical investigation of twenty software startups. In: Baumeister, H., Lichter, H., Riebisch, M. (eds.) Agile Processes in Software Engineering and Extreme Programming, pp. 20–36. Springer International, Cham (2017)
18. Dybå, T., Sjøberg, D.I.K., Cruzes, D.S.: What works for whom, where, when, and why? On the role of context in empirical software engineering. In: Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 19–28. (2012). https://doi.org/10.1145/2372251.2372256
19. Basili, V.R., Caldiera, G., Rombach, H.D.: Goal question metric paradigm. In: Marciniak, J.J. (ed.) Encyclopaedia of Software Engineering, vol. 1, pp. 528–532. Wiley, New York (1994)
20. Nguyen-Duc, A., Shah, S.M.A., Ambrahamsson, P.: Towards an early stage software startups evolution model. In: 2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 120–127 (2016). https://doi.org/10.1109/SEAA.2016.21

Software Startup ESSENCE: How Should Software Startups Work?

Kai-Kristian Kemell, Anh Nguyen-Duc, Xiaofeng Wang, Juhani Risku, and Pekka Abrahamsson

Abstract Software startups need to work in a systematic fashion just like mature organizations. However, existing software engineering methods and practices are not aimed at software startups. They do not account for the business aspect of startups and may not be well suited for software startups in general. The Lean Startup Methodology on the other hand contains some useful practices for software startups but is nonetheless impractical, offering little in the way of telling you what to do. Software startups are thus required to tailor their own method. Currently, many software startups simply work ad hoc or use various Agile methods and practices. In terms of Agile methods and practices, little consensus exists between startups. In this chapter, we discuss methods and method tailoring. We give guidelines on how to create your own way of working and recommend a tangible tool for doing so: the Essence Theory of Software Engineering.

Keywords Startup methodology · Startup essence · SEMAT essence · Startup gamification

K.-K. Kemell (✉) · J. Risku · P. Abrahamsson
University of Jyväskylä, Jyväskylä, Finland
e-mail: kai-kristian.o.kemell@jyu.fi; juhani.risku@jyu.fi; pekka.abrahamsson@jyu.fi

A. Nguyen-Duc
Department of Business and IT, University of Southeast Norway, Bø i Telemark, Norway
e-mail: angu@usn.no

X. Wang
Free University of Bozen-Bolzano, Bozen-Bolzano, Italy
e-mail: xiaofeng.wang@unibz.it

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_6

1 Introduction

Software startups should work in a systematic fashion in order to be effective, according to Eric Ries [1], a high-profile expert on startups and a startupper himself. In practice, working systematically means utilizing a method that is known and understood by the team. A method tells you how to work.

The issue here is that little exists in the way of methods for software startups. There are no widely known software engineering methods specifically aimed at software startups. While software startups can utilize the numerous existing software engineering practices and methods as they wish, they may not be well suited for software startups. Similarly, the Lean Startup Methodology [1] offers startups some ideas, principles, and practices to utilize, but does not directly tell you how to work and what to do. In other words, it is not practical. Each startup is thus responsible for creating its own method, or a way of working.

In this chapter, we will discuss how a way of working can be devised by a software startup and discuss The Essence Theory of Software Engineering as a tool for doing so. The goal of this chapter is to underline the importance of actively reflecting on your way of working and to provide some guidelines for doing so.

Although software startups have been extensively studied in the academia, little research currently exists on this point of view [2]. This chapter presents some preliminary results from an ongoing study in the form of a deck of practice cards presented in the second to last section. In addition, we provide some tools to utilize in this context based on past studies as well.

2 How Should Software Startups Work? Ideas and the Current State of Practice

A method is your way of working. It describes the way you work, should work, or want to work according to the description. A method can be broken down into parts that can be referred to as practices. Practices are microlevel methods that, again, describe some parts of the work process. Together, practices form methods. In terms of software engineering, a method is, e.g., Scrum[1] [3] while a practice is, e.g., pair programming. A (communication) practice can also be something as simple as using a specific software to communicate in a specific way inside the team.

Software startups develop software and can thus utilize various existing models, methods, and practices from the field of software engineering (SE). SE methods are numerous, ranging from the early so-called waterfall models to the currently fashionable, highly diverse Agile methods. Agile itself is not a method but a set of principles outlined in the Agile manifesto,[2] and the number of methods and practices that can be considered Agile is vast [4].

[1] https://www.scrum.org/

[2] https://agilemanifesto.org/

Existing SE methods are not specifically aimed at startups. In fact, on the contrary, they are generally intended for project use inside mature organizations. These SE methods thus focus entirely on SE, omitting the business aspect that is vital for startups. In these more mature organizations, the software developers are not interested in the work of the financial department and often have to care little for how the software is marketed toward its intended customers as that is also the job of another department. In a startup, on the other hand, the far smaller organization size often makes these roles more intertwined especially early on in the life of the software startup. Therefore, while these methods can still be utilized by startups, they can only help startups with their software engineering, which is only a portion of the work of a startupper.

In practice, software startups utilize a wide variety of methods and practices for developing software, based on a study by Paternoster et al. [5]. According to the study, software startups largely utilized various Agile methods and practices or simply worked ad hoc. Little consensus on what the best method or practice(s) for software engineering would be was found between the software startups studied by Paternoster et al. [5]. This further highlights the idea that startups, much like mature organizations, should concern themselves with finding or creating a method that suits their work in particular.

Aside from SE methods, there is little currently available in the form of more comprehensive methods for software startups. The lean startup, for example, is not a method that tells you how to work. It is more akin to the Agile manifesto in that it merely presents some principles associated with the idea of a lean startup. Some practices such as the Build–Measure–Learn loop are considered important or even vital for startups wishing to be lean, although they still offer little in terms of directly telling what to do.

Aside from, and including, the Build–Measure–Learn loop, various good practices recommended by different experts and startuppers alike do exist, even if they do not come bundled into any method(s). These startup practices can help you carry out some parts of your work better. Popular practices associated with startups include the aforementioned Build–Measure–Learn loop, the Five Whys, and the Minimum Viable Product. For example, the Five Whys practice is about understanding the root of a problem.

An example of this practice is presented as follows (taken from Wikipedia[3]):

Problem: The car would not start
- **Why?** The battery is dead.
- **Why?** The alternator is not functioning.
- **Why?** The alternator belt was broken.
- **Why?** The alternator belt was old and had not been replaced.
- **Why?** The vehicle had not been maintained properly.

While such practices are useful, they do not offer comprehensive tips or tricks for success. They do not offer roadmaps or tell you what to do next. They can direct your

[3] https://en.wikipedia.org/wiki/5_Whys—as it was on the 13th of March 2019.

way of working into the right direction, but it is up to you to make the most of them. They can be valuable when included into your method, but it is your job to create that method. Startups, just like mature organizations, should concern themselves with working in a systematic fashion in order to be effective [1].

3 How to Develop Your Method or Way of Working?

Whether you are working ad hoc, deciding on how to do certain things as new tasks arise, or you are utilizing a formal method like Scrum, you have a way of working. A way of working comprises all of your work practices from the way your team communicates to what software tools do you use to perform various tasks. Together, these various processes, practices, methods, and tools represent a way of working of a startup.

As we established in the previous section, there is currently no universal recipe for success for software startups. Software startups can utilize existing software engineering methods and practices, but they do not tackle the business aspect of a startup. Business-related methods for startups are scarce, and while various good practices recommended by practitioners exist (the Five Whys, etc.), they do not tell you how you should be working outside that one practice. These existing SE methods and various practices can, however, be utilized and combined in order to create your own method.

Your way of working encompasses everything you do in your startup, from your communication practices and your work hours to the tool(s) you use to develop software. How do you communicate: where and when do you have meetings, how do you communicate online when not in the office? Which tools do you use while developing your software?

Creating a way of working is not a one-time effort, as your way of working is not static. As new tasks arise and old ones are completed, what you work on is constantly changing. Thus, in order to work effectively, you should actively reflect on your way of working as you progress. Practices that were useful at one point may become less relevant and ultimately it might be better to phase them out completely. This is something that generally happens naturally, but it can be even more effective to do it consciously and to discuss it within the team.

Indeed, improving your way of working is highly related to consciously seeking to learn from what you are doing. Pay mind to what you are doing and why you are doing those things. Use metrics to measure how some of your practices work and then compare them to alternatives (we talk more about using metrics in the chapter titled “100+ Metrics for Software Startups—Common Practices of Using Metrics”).

Practical Example: You are starting a display advertising campaign. Rather than putting all of your eggs into one basket, you may wish to try different channels (e.g. Google, Facebook etc.) simultaneously. Track where the traffic is coming from. How much did you spend on each channel and how much did each channel bring in traffic in comparison? Pay mind to your advertising settings as well: whom are you targeting, which countries are you targeting, when are your ads showing for the people etc.

> Then, based on the data, pick the best channel(s) and focus on them. Try out two different ads in the same channel over the course of e.g. two weeks, one week for each. Which one brought in the most traffic? In this fashion, you can utilize metrics to decide which practice works best for your startup.

If you decide to utilize existing practices and methods, it is also worthwhile to evaluate them as you use them. You may find that modifying them rather than following them by the book works better for you.

4 Using the Essence Theory of Software Engineering to Describe and Improve Your Way of Working

The Essence Theory of Software Engineering [6] is a modular framework for constructing your own way of working. In short, Essence gives you a foundation that you can use as a foundation to construct your own way of working. This foundation comes in the form of a so-called kernel. The elements in this kernel can then be expanded upon using the Essence language in order to tailor your own way of working. Essence supports the use of any existing software engineering method or practice or combination of such [3].

4.1 Why Is This Relevant?

Essence provides you with a tool that can help you systematize your way of working [7]. Communication tends to be the most important problem in most projects and Essence is a tool meant to facilitate and improve communication. You may feel that modeling your way of working using Essence is stating the obvious that your team already knows. This is not the case at all, as the point of Essence is not to model our way of working for the sake of doing so for busywork.

By modeling your way of working using Essence you present it in a formal, visual manner. This can be thought-provoking and offers a good basis for discussion within the team. Does everyone agree that this is a good way to be doing things? If not, why? And how should it be changed?

Aside from using Essence to build and describe your way of working, Essence is useful for progress management [7]. The alphas and alpha states can help you better understand where you currently stand in terms of the software you are developing or your startup in general. Again, though it may initially feel like extra work to establish something you already know, your team may in fact not fully be on the same page about your progress.

Essence is not something one person in your team should utilize alone in order to manage your progress or to model your way of working. It is a communication tool and it should be used like one. Essence facilitates communication within the team and is helpful in communicating your progress and/or your way of working in a clear manner. It can also serve as a way to motivate your team members to think about their own work practices as individuals through your team meetings.

4.2 Quick Overview of Essence

The kernel is the foundation that contains the building blocks you can use to build your own method. It contains the basic elements present in every single software engineering project. The kernel consists of three different types of elements split into three categories called *Areas of Concern*. These areas of concern are denoted by three colors:

- *Customer* (Green)
- *Endeavor* (Yellow)
- *Team* (Blue)

These categories each contain three types of elements:

- *Alphas*, a.k.a. Things to Work With
- *Activity spaces*, a.k.a. Things to Do
- *Competencies*, i.e. the skills required to carry out tasks

The core of the kernel (seen in Fig. 1) is formed by the *alphas*, of which there are seven. These seven alphas are present in every software engineering project and thus they are the essentials. In other words: most projects have other things to take into account as well, but every project includes at least these elements among other things. The alphas are color coded according to the above split into areas of concern, also denoted in the kernel itself.

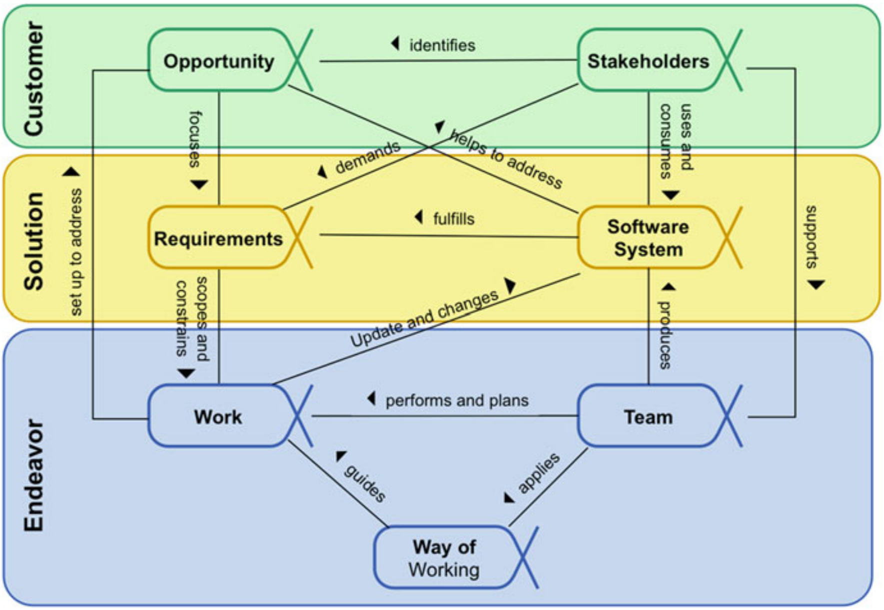

Fig. 1 The essence Kernel Alphas (http://semat.org/quick-reference-guide)

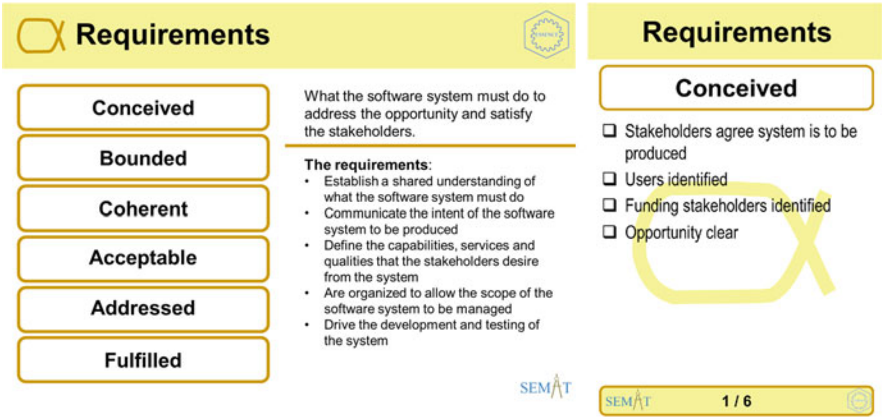

Fig. 2 The *requirements* alpha and its first state (http://semat.org/quick-reference-guide)

Each of the seven alphas has *alpha states* which are meant to help track progress in your startup. For example, the *requirements* alpha, below, starts with the requirements being outlined and ends when the software has fulfilled them (Fig. 2). Every software has requirements even if they do not directly come from a customer commissioning it.

In addition to the alphas, the kernel contains *activity spaces*, which contain some of the things that should be done in every software engineering project. As was the case with alphas, every project contains other things to do as well, but these are the essential things that should be done in every project. Finally, *competencies* are the skills required to carry out the tasks. For example, if you are developing a game using Unity, a developer joining your startup should probably be familiar with Unity, or at least be ready to learn it. The kernel contains some competencies that are required in every project.

The kernel contains only the essentials present in every project, but every project has its own elements aside from the essentials. For the kernel to be more useful for you, you can utilize the Essence language to add the unique characteristics of your software startup to it. Add your own alphas, describe your own practices, and draw your own method using the graphical elements of Essence.

4.3 Essentializing Your Way of Working

The act of describing your work practices using Essence is called essentializing them. This is done by using the Essence language to describe the practices. Essentializing a practice has three steps:

1. Identifying the elements. Use the element types of the Essence language to describe the elements present in the practice. The output is a diagram (see Fig. 3).

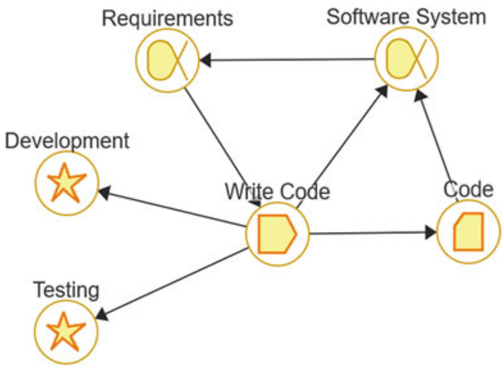

Fig. 3 Pair programming described with Essence, drawn with Essencery (Essencery.com)

2. Drafting the relationships between the elements. Make practice cards.
3. Providing further details. Add descriptions, references, etc. to the cards.

In addition to the alphas, activity spaces, and competencies discussed in the previous subchapter, the Essence language contains a few other elements. These are *activities*, *work products*, and *patterns*. Activities, like activity spaces, are things to do, but activity spaces are placeholders and can act as containers or umbrellas for groups of activities. Work products are tangible things you create as you work, e.g., meeting notes or PowerPoint presentations about your software idea. Finally, patterns are arrangements of other elements presented in the language, such as roles (programmer, designer, etc.).

How much you conform to the Essence language as you essentialize your practices and describe your way of working is up to you. Ultimately, it is not necessarily important that you are fully conformant to the language. Essence is a communication tool and if what you are doing facilitates communication in your team while successfully describing what you wish to describe with the language, you have reached your goal.

You do not have to precisely tie the practices and practice cards to, e.g., the alpha states, especially at first. Once you have devised a method that really works for your team, one that you can keep using in the future, you may consider doing so. Being fully conformant with Essence takes time and if you are continuously changing your way of working, it may be better to settle for a lower conformance level in your practice descriptions. You do not have to utilize Essence in its entirety, by the book, for it to help you. In the next subsection, we will provide you with links and resources that can help you get started with Essence.

4.4 Getting Started with Essence

If you are interested in Essence, there are various resources available online that can give you a more in-depth overview of it, as well as tools that can help you get started with it more easily. Below is a list of resources that you can utilize to get started with Essence:

- *OMG Standard for Essence*. The full documentation for Essence[4] [8].
- *Quick Reference Guide*. A formal guide for getting started with Essence.[5]
- *SematAcc*. A digital tool for tracking alpha states[6] [9].
- *Essencery*. A digital tool for drawing graphs using the Essence language[7] [10].
- *Ivar Jacobson Practice Card Library*. A database of various existing software engineering practices made into Essence practice cards. You can get ideas for new practices from here and it can save you lots of time when making practice cards.[8]
- *Essence of Software Engineering—The Board Game*. A board game to teach you the idea of Essence[9] [11].

4.5 Essentials for Early-Stage Software Startups: An Example of Using Essence

Using Essence as a base, we have developed a set of practices intended to help early-stage software startups (Fig. 4). Though later on startups branch out and become very different, very early on in their lives they share more similarities. Building on these similarities, we have devised a set of practice cards for use with Essence.

These cards are intended to guide an early-stage software startup from idea generation to the early stages of developing their first product. Each card describes a practice that is intended to help early-stage startups progress. As an early-stage startup, the deck can:

- Offer you an additional starting point for essentializing your way of working
- Give you ideas on what to do next
- Give you ideas for new practices

This set of cards was developed for a university course and as a part of an ongoing study. The deck is being improved iteratively and this is the result of the first iteration, used as a teaching tool in a course teaching startup entrepreneurship at the University of Jyväskylä. These cards have not yet been formally bound to the Essence kernel and act as a stand-alone tool as well. There are 12 cards in total in this deck.

As these cards were developed for a university course on Software Startups, some of the cards contain lingo indicating this original use purpose. These cards are available in their entirety on FigShare (https://doi.org/10.6084/m9.figshare.8378900.v1).

---

[4] https://www.omg.org/spec/Essence/

[5] http://semat.org/quick-reference-guide

[6] https://ineed.coffee/projects/sematacc-2/—can be downloaded

[7] Essencery.com—used directly in the browser

[8] https://practicelibrary.ivarjacobson.com/start—requires registration

[9] https://doi.org/10.6084/m9.figshare.8052653.v1—permanent link to all the game materials

Fig. 4 The card deck

5 Methodology: Creating the Card Deck

The card deck presented in the preceding section (Sect. 4.5) was created as a part of an ongoing study on Essence in the context of software startups. The idea of the study is to ultimately tailor a version of the Essence kernel for software startups. In the process, we aim to create a practice card deck for early-stage software startups. The first iteration of the said practice card deck was featured in this chapter.

The first iteration of the card deck was based on existing academic and practitioner literature on software startups. For example, as a basis for the deck, we used the life cycle stages discussed by Wang et al. [12] in their study. According to this view, a software startup begins with problem definition as they come up with a problem to solve, or a need to address. They then move on to problem validation as they try to ascertain that the problem or need is in fact a real one someone wants to have addressed and is perhaps willing to pay to have addressed. Having validated the problem, the next stage is to then come up with a solution to carry out the idea: solution definition. Once the solution has been defined, it, too, must be validated. The solution is validated in practice when (if) the startup becomes a functioning firm, at which point the software startup slowly grows into a mature organization and ceases to be a startup.

Building on this idea of four life cycle stages, we devised a card deck depicting key practices associated with this process. These practices are well-established ones discussed in academic literature and by experts alike, such as pivoting or validation using an MVP. These cards are still being developed and thus more practices are likely to be added in future versions.

This iteration of the deck was deployed in a university course on software startup entrepreneurship at the University of Jyväskylä. In the course, teams of 3–5 students were to found a software startup based on a new or existing idea. We gave each team a physical deck of the cards to use. The cards were intended to guide them by highlighting various actions they should perform during the course, and more importantly, to progress as startups. Data was collected on their experiences with the cards by having the teams report their use of the cards by writing on them. This data will be used to improve the cards for future iterations.

Finally, it should be noted that the cards were devised with the educational setting in mind. Thus, some of the cards contain some material that may not be fully relevant to a software startup out on the field. For example, the "Startup Spirit" card discusses treating the startup as a real startup and not simply a course project, which is something that does not directly concern most startups.

Key takeaway

- ✓ Startups, just like more mature organizations, should work systematically.
- ✓ Model your way of working in order to critically evaluate it.
- ✓ Continuously pay mind to how you work and look for points of improvement.
- ✓ Discuss your way of working with your team in order to improve it.
- ✓ Use metrics to objectively evaluate your work practices.
- ✓ You can use Essence to model your way of working.
- ✓ If you are interested in Essence, use the tools suggested in 4.4 to get started.
- ✓ The deck of cards we presented can give an early-stage software startup new ideas for work practices and can be used as a base to build a way of working on using Essence.

6 Conclusions: Managerial Implications

In this chapter, we have discussed the importance of working in a systematic fashion and constantly improving your way of working. Though software startups should also seek to work in a systematic fashion, methods and practices aimed at software startups are rare in software engineering. Existing methods are devised with large, established organizations in mind. They are aimed at project use and focus exclusively on the software engineering aspect of the project, leaving out the business aspect that is just as important for a software startup.

Similarly, little exists in the way of startup-oriented business practices or methods. Though some practices recommended for startups such as the Five Whys and the Build–Measure–Learn loop discussed in the context of the Lean Startup methodology exist, even the lean startup methodology does not offer startups actionable and practical advice on how to work. Thus, it is up to each software startup to devise its own method.

In this chapter, we discussed how a method or way of working should be devised, and recommended Essence as one tool for doing so. Though no comprehensive methodologies for software startups exist, startups should nonetheless aim to work in a systematic fashion [1]. In devising their own method, software startups can utilize these various existing practices in creating a method by using them as building blocks. Existing methods such as Scrum can be tailored in this fashion to better suit the context and needs of a startup as opposed to a large, established organization.

Once a method is created, it should be iteratively developed. This can be done by, e.g., measuring the effectiveness of individual work practices, as we discussed in Sect. 3. The method, and the practices it consists of, should be discussed within the team in order to make sure everyone is on the same page. This way, everyone can provide their point of view on the way the team works—or should work.

References

1. Ries, E.: The Lean Startups: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Books, New York (2011)
2. Unterkalmsteiner, M., Abrahamsson, P., Wang, X.F., Nguyen-Duc, A., Shah, S., Bajwa, S.S., Baltes, G.H., Conboy, K., Cullina, E., Dennehy, D., Edison, H., Fernandez-Sanchez, C., Garbajosa, J., Gorschek, T., Klotins, E., Hokkanen, L., Kon, F., Lunesu, I., Marchesi, M., Morgan, L., Oivo, M., Selig, C., Seppänen, P., Sweetman, R., Tyrväinen, P., Ungerer, C., Yagüe, A.: Software startups – a research agenda. e-Inf. Softw. Eng. J. **10**(1), 89–123 (2016)
3. Park, J.S., McMahon, P.E., Myburgh, B.: Scrum powered by essence. ACM SIGSOFT Softw. Eng. Notes. **41**(1), 1–8 (2016)
4. Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J.: Agile Software Development Methods: Review and Analysis, p. 478. VTT, Otamedia (2002)
5. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: a systematic mapping study. Inf. Softw. Technol. **56**, 1200–1218 (2014)

6. Jacobson, I., Ng, P., McMahon, P.E., Spence, I., Lidman, S.: The essence of software engineering: the SEMAT kernel. ACMQueue. **10**, 40–52 (2012)
7. Kemell, K.K., Nguyen-Duc, A., Wang, X., Risku, J., Abrahamsson, P.: The essence theory of software engineering – large-scale classroom experiences from 450+ software engineering BSc students. In: Proceedings of the 2018 International Conference on Product-Focused Software Process Improvement (PROFES 2018) (2018)
8. Object Management Group: Essence – Kernel and Language for Software Engineering Methods. Version 1.1. http://semat.org/essence-1.1. Accessed 28 May 2018
9. Graziotin, D., Abrahamsson, P.: A web-based modeling tool for the SEMAT essence theory of software engineering. J. Open Res. Softw. **1**, e4 (2013)
10. Evensen, A., Kemell, K., Wang, X., Risku, J., Abrahamsson, P.: Essencery – A Tool for Essentializing Software Engineering Practices. arXiv preprint (2018). https://arxiv.org/abs/1808.02723?context=cs
11. Kemell, K.K., Risku, J., Evensen, A., Dahl, A.M., Grytten, L., Jedryszek, A., Rostrup, P., Nguyen-Duc, A., Abrahamsson, P.: Gamifying the escape from the engineering method prison – an innovative board game to teach the essence theory to future project managers and software engineers. In: Proceedings of the 2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), Stuttgart (2018)
12. Wang, X., Edison, H., Bajwa, S.S., Giardino, C., Abrahamsson, P.: Key challenges in software startups across life cycle stages. In: Sharp, H., Hall, T. (eds.) Agile Processes, in Software Engineering, and Extreme Programming. XP 2016. Lecture Notes in Business Information Processing, vol. 251. Springer, Cham (2016)

Startup Metrics That Tech Entrepreneurs Need to Know

Kai-Kristian Kemell, Xiaofeng Wang, Anh Nguyen-Duc, Jason Grendus, Tuure Tuunanen, and Pekka Abrahamsson

Abstract Metrics can be used by firms to make more objective decisions based on data. Software startups in particular are characterized by the uncertain or even chaotic nature of the contexts in which they operate. Using data in the form of metrics can help software startups to make the right decisions amid uncertainty and limited resources. However, whereas conventional business metrics and software metrics have been studied in the past, metrics in the specific context of software startups have not been studied. In this chapter, we present the results of a multivocal literature review to offer you 118 metrics practitioner experts think software startups should measure. These metrics can give you ideas for what your startup should measure.

Keywords Software metrics · Startup metrics · KPI · Data-driven startup

1 Introduction

Metrics are data that can be used to measure objects or phenomena. We use various metrics every day, from weighing objects at the store to driving at certain speeds with our cars. Even seemingly qualitative data can be made into metrics by quantifying it. For example, so-called Likert scale surveys are used to convert

K.-K. Kemell (✉) · T. Tuunanen · P. Abrahamsson
University of Jyväskylä, Jyväskylä, Finland
e-mail: kai-kristian.o.kemell@jyu.fi; tuure.t.tuunanen@jyu.fi; pekka.abrahamsson@jyu.fi

X. Wang
Free University of Bozen-Bolzano, Bozen-Bolzano, Italy
e-mail: xiaofeng.wang@unibz.it

A. Nguyen-Duc
Department of Business and IT, University of Southeast Norway, Bø i Telemark, Norway
e-mail: angu@usn.no

J. Grendus
3D Ventures Oy, Singapore, Singapore

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_7

opinions into numerical values by presenting the responders with statements they are to either agree or disagree with using a numerical scale.

However, while data is valuable in supporting decision-making, in the everyday practice manager intuition is still just as important for strategic decision-making in organizations [1]. We argue that though manager intuition can be powerful, using data to support it can make it even more so. This has been emphasized by the rising value of data in the past few decades. Firms across industries have taken an interest in collecting and analyzing data in order to improve their business based on, e.g., customer behavior on their website.

The idea of big data [2], a term coined in the 2010s, further highlighted the importance of data. The big data discourse urged companies to gather any and all data even if they did not necessarily have any use for it right there and then. These vast amounts of data, then, could be analyzed later in search of various correlations, especially out-of-the-box ones that may not have ever occurred to anyone without access to such large amounts of data.

Software startups, just like mature organizations, can utilize metrics to make decisions or to track progress. As they typically operate under a lack of resources and in particularly uncertain contexts [3], metrics can help startups make the right decisions at the right time. However, based on past survey data, we discovered that half of the responding 4000+ software startups in fact did not use metrics at all. Forty-one percent of the startups felt that it was too early for them to use metrics. Out of the remaining 59%, 16% felt that they did not have the resources to track metrics or that doing so would not benefit them, and another 14% tracked them but reported that they did not use them to support decision-making.

Motivated by this data, we sought to give software startups ideas for what metrics they could use. In this chapter, **we present a list of 100+ metrics for Software Startups that we compiled by studying practitioner-oriented literature** (e.g., online blogs, articles interviewing experts) discussing metrics for software startups.

2 Expert Opinions: What Should Software Startups Measure?

In utilizing metrics, software startups combine various types of metrics. They can utilize conventional business metrics, as well as business metrics more specifically aimed at startups, as well as software-related metrics including website metrics. Across different life cycle stages (e.g., those proposed by Wang et al. [4]), different metrics can be important for software startups. For example, conventional financial metrics are not as relevant for early-stage startups that may still be in the process of acquiring their first customers or that are still calculatingly running a deficit for the time being. A more relevant metric in such a situation could be to simply measure the amount of remaining expendable capital.

Through a multivocal literature review (described in second to last section of this chapter), we compiled an extensive list of 118 metrics for software startups. Much of the literature reviewed for this study consisted of short “n metrics your startup should measure” types of lists. As a result, there was a considerable amount of overlap in the results. This does, however, point to there being some consensus among practitioner experts as to which metrics are particularly interesting or important.

These 118 metrics are divided into three separate categories in this section: business metrics, user and customer metrics, and software engineering metrics. Each category will be presented separately and discussed in detail in the following subsections.

2.1 Business and Financial Metrics

Business and financial metrics here refer to metrics-related costs and revenues. Business and financial metrics are largely metrics that communicate the current situation of the business using various monetary indicators. The business and financial metrics presented here are rather universal metrics used by small and large businesses alike, as well as startups. However, metrics that could be considered to be business metrics but which are closely related to the users or customers (e.g., average lifetime value of the users) are considered to more importantly be user and customer metrics, which are discussed and presented in the following subsection (Table 1).

2.2 User and Customer Metrics

Understanding the users of a software is vital for a software company, just as understanding one’s customers is vital for any business. Digital services offer unique opportunities for businesses to collect data from their users which can then help them better understand their needs. Instead of having to actively consult the users or customers, digital services make it possible to observe and track them as they use the service, making it possible to collect large amounts of data from the users without having to ask them anything.

For example, Bounce Rate refers to the percentage of the visitors of a website that leave the site without performing any actions on it. It does not directly help one understand why they are leaving, but it can indicate that something is wrong with the website. Recording historical bounce rates, then, makes it possible to see if changing the website in various ways has a positive effect on the bounce rate—or not. Such metrics make it possible to understand users to certain extents even without directly communicating with them (Table 2).

Table 1 Business and financial metrics recommended for software startups

Metric (and up to 3 references)	Description of the metric
Abandonment [5]	Transactions abandoned before completion
Ad Inventory [5]	Total views of each ad in a time period
Ad Rates [5]	Value of each ad inventory
Annual Contract Value [6–8]	Avg. annualized revenue per customer contract
Annual Recurring Revenue [6, 8, 9]	Predictable revenue annually (e.g., subscriptions)
Annual Run Rate [6]	Projected annualization of monthly recurring revenue
Avg. Revenue per User [6, 10, 11]	Avg. revenue per user over a time period
Avg. Revenue per Customer [6, 7, 11]	Avg. revenue per customer over a time period
Billings [6]	Current quarter revenue plus deferred revenue from the previous quarter
Breakeven Analysis [12]	Analysis to determine the point where revenue covers the costs of receiving it
Burn Rate [10, 13, 14]	Rate at which available capital is used
Campaign Contribution [5]	Added revenue from an ad campaign
Capital Raised to Date [15]	Amount of investment capital raised in total
Cash Flow Forecast [12]	Forecast of financial liquidity in a period of time
Cash on Hand [16]	Available capital
Committed Weekly Recurring Gross Profit [17]	Percentage increase in profits weekly committed recurring profit
Compounded Monthly Growth Rate [6]	Avg. % growth per month since inception, or another start point for measuring
Cost of Goods Sold [15]	Cost of products or services sold (e.g., hosting)
Customer Acquisition Cost [12, 13, 18]	Average cost of acquiring a paying user
Customer Acquisition Cost to Lifetime Value Ratio [19, 20]	Customer Acquisition Cost vs. Customer Lifetime Value
Customer Concentration [6, 21]	Revenue from largest customer vs. total revenue
Customer Retention Cost [11]	Amount of spending on customer retention
Deferred Revenue [6]	Revenue received in advance of earning it
Fixed vs. Variable Costs [12]	A measure of total spending split by source
Gross (Cash) Burn [6]	Monthly expenses and any other outlays
Gross Margin [6, 10, 18]	Total revenue compared to cost of goods sold
Gross Profit [6–8]	Total revenue minus cost of goods sold
Market Share [22]	
Market Value [22]	
Monthly Cash Burn Rate [6, 20]	
Monthly Recurring Revenue [6, 19, 23]	Monthly predictable revenue (e.g., subscriptions)
Month-on-Month Growth [6, 7, 23]	Average of monthly growth rates

(continued)

Table 1 (continued)

Metric (and up to 3 references)	Description of the metric
Net (Cash) Burn Rate [6]	Gross cash burn vs. revenue in a period of time
Number of Transactions [24]	Number of transactions made in a time period
Operation Efficiency [10, 14]	Comparison of firm expenses by source
Payback Time [11]	Time to recoup from an expense via revenue
Payment Failures [17]	Number of failed transactions from users
Platform Risk [6]	Dependence on a specific platform or channel
Profit Margin [7, 11, 20]	Revenue minus cost divided by revenue for a product. Different ways to measure for, e.g., Software-as-a-Service companies
Return on Advertisement Spending [18]	Profits divided by advertisement spending
Revenue [7, 8, 25]	Total revenue
Revenue Growth Rate [9, 26]	
Revenue Run Rate [10, 19]	
Sell-Through Rate [6]	No. of units sold in a time period in relation to the no. of items in inventory at its beginning
Time to Customer Breakeven [5, 20]	Time it takes to recoup from Customer Acquisition Cost
Total Addressable Market [6, 7, 22]	Total hypothetical market size
Total Contract Value [6–8]	Value of one-time and recurring charges

2.3 *Software Engineering Metrics*

SE metrics, as discussed in the second section, comprise both process and product (or service) metrics. They thus include metrics both related to the development process carried out by the organization responsible for the development, as well as metrics related to the software in its operational life. They also include website metrics that are not considered user and customer metrics first and foremost (Table 3).

2.4 *Social Media Metrics*

Social media platforms such as Facebook are important for businesses looking to grow their visibility. It enables them to interact with users or customers and potential users through the platforms. For software startups, the way social media offers a cost-free way of potentially boosting the visibility of their service is highly valuable. Though it is possible to purchase paid advertisements on many of these platforms, it is equally possible to simply use the platforms to gain visibility free of charge by creating content on the startup's page on each platform.

Though much of the potential success hinges on how interesting the posts of the company are seen as being by the general public on the platform, metrics can help

Table 2 User and customer metrics recommended for software startups

Metric (and up to 3 references)	Description of the metric
Acceptance Rate [5]	Avg. no. of invites accepted by new users
Activation Rate [6, 11, 13]	Number of visitors or users performing a specific action such as registering or installing
Active User Growth Rate [5]	No. of new active users in a time period
Average Time on Hold [5]	Time user spends on hold when calling support
Bounce Rate [13, 27]	Percentage of visitors leaving website quickly
Churn Rate [7, 10, 28]	Lost users or customers over a time period
Click-Through Rate [5]	Visitors that clicked a specific website link
Content Creation [5]	No. of visitors that interact with website content
Conversion Rate [7, 13, 28]	No. of visitors that become users or customers, or no. of users that become customers.
Customer Count [24]	Total number of customers (paying users)
Daily Active Users [6, 19, 29]	No. of users who use the software daily
Daily Active Users to Monthly Active Users ratio [11]	A more detailed measure of user activity
Direct Traffic [6]	Traffic coming in directly
E-mail Conversion Rate [30]	Number of recipients that, e.g., became users
E-mail Open Rate [30]	No. of mailing list members that open an e-mail
Frequency of Logins [7]	Average frequency of user logins
Frequency of Visits [11]	Average frequency of visits to, e.g., website
Gross Churn Rate [6, 31]	Total users lost
Intent to Use [30, 32]	Data indicating that a new user is about to start using the service, e.g., they imported custom data
Invitation Rate [5]	Avg. no. of invites sent per existing user
Launch Rate [5]	No. of downloaders that launched the software
Leads [33]	An estimate of prospective customers
Lead-to-Customer rate [33]	Number of leads converted into customers
Lifetime Value [12, 13, 18]	The average total revenue a customer generates
Monthly Active Users [13, 19, 29]	No. of users who use the software monthly
Monthly Churn Rate [6]	Lost users or customers per in a month
Net Adds [5]	Total new customers vs. cancelations
Net Churn [6]	New users gained vs. users lost
Net Promoter Score [6, 7, 29]	How likely users are to recommend service (survey)
Network Effects [6]	Effect of one user on the value experienced by other users (e.g., Metcalfe's Law)
New Visitors [7]	Number of new visitors
Number of Logins [6, 25]	Logins per user over a period of time
Organic Traffic [6]	Unpaid traffic from, e.g., Google search results
Prospects [5]	Number of users that might become customers
Purchases [5]	No. of purchases made by a user in a time period
Recency [34]	Days since last visit of user
Referrals from Current Users [13, 21, 35]	How often current users refer new users

(continued)

Table 2 (continued)

Metric (and up to 3 references)	Description of the metric
Referral Rate [28]	Volume of referred users or purchases
Registered Users [7]	Total number of registered users
Repurchase Rate [15]	No. of customers that made a purchase during the previous and current period of time
Retention Rate [13, 18, 28]	Percentage of users or customers still using the service after a period of time
Retention by Cohort [6]	% of original user base still using the software or conducting transactions in it
Reviews Considered Helpful [5]	Number of reviews considered helpful
Reviews Written [5]	Number of reviews written
Session Interval [7]	Average time between software use sessions
Session Length [7]	Length of average software use session
Sources of Traffic [7, 21, 35]	Source and volume of user traffic per source
Time to First Purchase [5]	Avg. time users take to become customers
Total Ad Clicks [5]	Number of advertisements clicked by visitors
Total Number of Customers [13, 36]	
Total Number of Users [22, 25]	Based on, e.g., registered user accounts
Traffic [14, 25, 28]	Total number of website visits (nonunique)
Traffic-to-Leads [28]	Total traffic in relation to potential customers
User Acquisition Rate [25, 29]	Total new nonpaying users in a time period
User Demographics [25, 29]	Avg. age, gender distribution, location, etc.
User Engagement [7, 29, 32]	Measured through, e.g., login frequency. Definition depends on context.
Unique Visitors [19]	Unique website visitors during a time period
Viral Coefficient [6, 19, 36]	No. of new customers each existing one converts

Table 3 Software engineering metrics recommended for software startups

Metric (and up to 3 references)	Description of the metric
Development Time [14, 24]	Time it takes to implement a new feature
Downloads or Installs [8]	Total amount of downloads or installs
Innovation Metabolism [37]	No. of completed build–measure–learn cycles
Load Time [29]	Time it takes for software to start or respond to user commands
Office Morale [25]	How motivated the team is (diff. metrics)
Stability [29]	Frequency of crashes in software use
Top Keywords Driving Traffic to You [5]	Search terms used by visitors to find your site
Top Search Terms [5]	Both those that lead to revenue and those that do not have any results
Uptime [27]	Percentage of time software or website is available and operational

Table 4 Social media metrics recommended for software startups

Metric (and up to 3 sources)	Description of the metric
Amplification Rate [11]	Number of shares on social media per customer
Facebook Likes [25]	Number of likes on firm Facebook page
Likes per Post [30]	Likes per social media post (or shares, etc.)
Social Media Reach [30]	Post reach within, e.g., Twitter or Facebook

you understand how successful your content there is. However, the experts did not place much emphasis on social media metrics in the literature we reviewed. Below are the metrics they discussed and recommended (Table 4).

3 How to Utilize These Metrics?

Any single metric will never solve all problems in all startups, and no single metric is as useful to every startup. However, tracking metrics helps you gain valuable data that you can then use to make decisions. Simply looking at your revenue does tell you something about how you are doing financially, but it will not tell you why. Furthermore, when used in unison, a combination of various metrics can tell you far more than any one metric alone can.

For example, if you wish to understand how many users you really have, looking at the amount of total registered users you have only scrapes the surface of the truth. You can then measure how many (unique) Monthly Active Users (MAU) you have to gain an understanding of how many of the registered users are actually still active. By then measuring Daily Active Users (DAU) you get more detailed information about how active the still active users really are. You can combine this with Cohort Analysis by splitting your users into groups based on when they initially created their accounts or began to use the service and how many of the users who registered more than a year ago are still active.

Having a basic understanding of your user activity, you can go into further detail by using more general-purpose user activity metrics. By looking at how often the users log in or use the software (session intervals), you can see how many times per month the Monthly Active Users log in. While at it, you can track the duration of their session lengths and how long do they stay on your site or play your game.

Time spent logged in does not tell the whole story either. Do they actually do the things you want them to do (user engagement)? Track their actions in your service while they are logged in. If they are not doing the things you want them to do, maybe something needs to be changed in your software? You could even approach the users who do not perform certain actions with a brief survey upon logout.

In this fashion, you can use combinations of metrics to gain a deeper understanding of some aspect of your business. However, even single metrics can provide valuable insights. For example, merely by tracking Daily Active Users you are able to see how many users are logging in every day. If the number suddenly starts dropping the day after an update was deployed, something was likely wrong with the update. Perhaps the update made the software unstable on some operating systems. Had you not been tracking your user activity through DAU or other such metrics (or been receiving error reports from the potential crashes), you may have only noticed the problem as a stark drop in revenue at the end of the month.

Finally, it can be beneficial to understand what metrics help you better your business and what metrics your stakeholders are interested in. Metrics that are important for your software and your team may not be interesting to potential investors who are mostly concerned with being able to see that you can become a viable business (if you are not already). Thus, you may wish to consider utilizing different metrics for internal use and for communicating with stakeholders. As you grow, the different people in your startup may also become interested in completely different metrics related to their job in your startup. Metrics do not have to be organization wide.

3.1 Using Metrics: A Practical Example

Various real-world stories depicting metric usage in startups exist online, from books to blogs. Whereas the previous example was a theoretical one, we cite a blog article by Sindre Hopland on the itnig blog[1] in giving you a practical one from a startup called Quipu. Quipu is a B2B (Business to Business) software startup providing a billing Software-as-a-Service (SaaS) solution.

The Quipu team considered churn rate as one of their key metrics indicating their performance. The less customers a SaaS solution loses over time, the better, and for a B2B one, acquiring new ones is even more expensive than for a B2C (Business to Customer) software startup. However, in practice, a low churn rate was simply their target. The team utilized various other metrics to help keep the churn rates low.

In the case of Quipu, the CEO considered customer support to have been the most important way of keeping their churn rate in check. Early on, he handled it by himself. By dealing with customers hands-on, he was able to understand what they were having problems with. Additionally, he was able to discuss the product with them and ask for any improvements they might wish to see. This can be a valuable way of conducting user interviews that requires little setting up.

While data gained in this fashion is typically qualitative, understanding your users is always valuable. Another valuable source of qualitative data for the Quipu team was their unsubscribe feature. Upon cancelation, the team would be notified and would contact the customer before finalizing the cancelation. In some cases, the customer was experiencing problems the team was then able to solve, keeping the customer.

[1] https://blog.itnig.net/2017/01/15/how-quipu-kept-churn-rates-under-one-percent-ever-since-their-launch/

Aside from providing users, support when they ask for it, Quipu actively approached their customers based on various metrics. As they provided a SaaS solution, they were able to utilize various metrics discussed in this section to track the actions of their users while they used the service. In doing so, they were able to preemptively contact customers who were struggling with their service. Using this data, they were also able to establish patterns. For example, a user who created three invoices in their system was much more likely to stay as a customer.

This highlights an important lesson: it can be worthwhile to adopt the idea of big data even early on. Even if you have no immediate use for the data you are collecting, you can study it in retrospect to discover patterns that may turn out to be helpful for your company. Furthermore, this example highlights the way metrics are best used in conjunction. Though the churn rate was what they company was concerned with, they utilized a number of metrics and gathered various types of data to actually control their churn rate.

4 Related Academic Work: Software Startups and Metrics

Startups are studied across disciplines, primarily in various economic disciplines and in information technology ones. However, in software engineering we speak of software startups, many practitioners simply speak of *startups* while in practice referring to software startups. By definition, software startups are startups that use software to *deliver* value. A software startup does not have to sell software or even focus on developing software to be a software startup. Uber, for example, delivers its value through its software application and is thus a software startup. While not all startups are indeed software startups even using this definition, the majority of them nonetheless are. Furthermore, economic disciplines commonly refer to (software) startups as New Technology-Based Firms (NTBF) [38].

From the point of view of metrics, software startups are not entirely unique. Rather, metrics that can be considered relevant for software startups come from various different areas of metrics. Software startups can utilize common financial metrics such as Net Present Value (NPV) or Return on Investment (ROI) when communicating with their stakeholders. Similarly, they can look at typical software or website metrics such as uptime to determine the status of their service.

As software startups do, however, differ from mature organizations, metrics that are important to more mature organizations may not be relevant at all for an early-stage startup. Even software startups are highly likely to find different metrics relevant in different stages of their life cycles, according to the stages proposed by Wang et al. [4]. After all, if you have no revenue yet, using revenue-related metrics makes no difference to you, although you can use them to make and communicate future revenue forecasts. In the following subsections, we will briefly discuss different categories of metrics that can be relevant for software startups, and which have been extensively studied in the past in various contexts and across scientific disciplines (e.g., [39]).

4.1 Software Engineering Metrics

Software startups can utilize generic Software Engineering (SE) metrics. More specifically, these SE metrics can be divided into product and process metrics [40]. Product metrics comprise metrics related to the software (e.g., uptime, load times) either during development or during its operational life. On the other hand, process metrics refer to metrics related to developing the software (e.g., development sprint duration) and are related to the development and operations team and their way of working [39].

Software Engineering metrics can also be considered to include *website metrics* as websites are ultimately software [40]. For businesses whose product (service) is not directly accessed through their website as SaaS, website metrics can offer different insights into their business. In such cases, website metrics are also clearly separate from product or service metrics. For example, website metrics can be used to determine conversion rates: how many visitors ultimately end up downloading the product or registering on the website.

Conventional website metrics such as site uptime or bandwidth [41] are still relevant, even though assuring site uptime even during heavy traffic has become easier in the wake of technological progress. Consequently, the focus on website metrics has shifted more toward understanding how the users interact with the website [42].

Finally, it is worth noting that few Software Engineering metrics related to the process side of SE metrics were discussed in the literature reviewed for this study. This is likely a result from the fact that SE methods and practices utilized by startups and mature businesses are highly diverse [43], from ad hoc methods to various in-house methods and by-the-book methods [44]. Different methods such as Scrum use different in-built metrics (sprint duration, etc.) and while these metrics are certainly applicable to any software startup utilizing these methods, they are not necessarily applicable to software startups not using that particular method. As such, if you are using a commonly used SE methodology or practice(s), you should consider looking into any metrics that are recommended to be used for measuring progress while using that method or practice.

4.2 Business and Financial Metrics

Though software startups differ from mature businesses, they can nonetheless still utilize various established business metrics. Business metrics are largely related to costs and revenues. By keeping track of your cost and revenue structure, you are able to have a better understanding of how your business is doing, and if costs need to be cut to stop making a loss, you know where to start. These types of metrics are also of interest to investors who want to see whether you can make a profit and keep growing your business.

However, especially early-stage startups are different from mature organizations in this regard as well. A newly founded startup may not even have any revenue to speak of yet. Thus, they might be far more interested in measuring their Cash Burn Rate and Cash-on-Hand to understand their current situation, as opposed to conventional business metrics such as Net Present Value [45]. They may utilize business metrics to make forecasts, though, and many investors are indeed interested in seeing estimates for, e.g., future revenue streams and how they can change based on different potential pricing models for the startup's service.

4.3 Usability Testing, Usability Metrics, and User Experience

Involving the user in the design of software was something advocated for in the Agile Manifesto when Agile Software Development (ASD) began to take root in the late 1990s. By involving the user, it was argued that it would be possible to better understand what they want, and it would, for example, make it possible to change the software during development to better match user needs as they emerge. This can be highly beneficial as it is much cheaper to make larger changes to a software service during development than during production or operation.

In practice, this typically means having a potential user test some version or a prototype of a software while either being observed or by self-reporting their use. There are different protocols for how the tests should be carried out, but generally the idea is to have a user use the software independently. If they struggle to get something done using the software, the observer, if one is involved in the test, will then typically step in to help them so that the test can continue without causing too much frustration to the subject. In this fashion, it is possible to see how people interact with the software in practice and what kind of difficulties they face while using it. They may ask for features that do not presently exist or they may end up being confused about something that you felt was very intuitive.

This is still a valid approach used today, although recent developments have seen the user focus shift toward indirect user involvement as opposed to direct user involvement. Rather than asking users anything directly, many software companies choose to collect data of the users of their software while they use the software and make changes on their software based on how the users use it. While this is a much less resource-intensive approach to understanding one's users, understanding the why behind the actions of the users requires more effort as you cannot ask them directly.

Usability testing has been widely studied in the academia and various guidebooks for involving the user into product design exist. In terms of usability testing, startups can arguably use the same methods as more mature organizations and thus we do not discuss them in depth here. If you are curious to read more about usability testing, Jakob Nielsen, for example, is a high-profile author for website usability.

Although these categories of metrics are well established and have been studied and used extensively in the past, little literature discussing these metrics from the point of view of startups exists. In this chapter, we thus discuss them from the point of view of startups by presenting the opinions of various practitioner experts on what startups should measure.

5 Methodology: How Did We Gather This Data?

The metrics presented in this chapter were compiled through a multivocal literature review of mainly practitioner literature. We conducted the literature review in three steps: (1) we reviewed literature written by high-profile experts (e.g., Eric Ries); (2) we carried out Google searches to find opinions of less high-profile experts; and (3) we reviewed sources the authors mentioned or cited in their texts.

For the Google searches, we followed a protocol in order to conduct the review in a systematic fashion (as opposed to, so to say, just googling around). We used the following search queries to find literature: "software startup metrics," "startup metrics," "startup metrics list," and "startup what to measure." Out of the results retrieved in this fashion, only hits that fulfilled the following criteria were included into this study:

- Not a clear advertisement (e.g., a firm writing a blog post to recommend their own data analytics tool)
- No unclear metrics or vague groups of metrics (e.g., "look at your sales")
- Text documents only (no videos, etc.)
- Only stand-alone documents written under real names (e.g., no Reddit posts)
- Only publicly available documents; not behind a pay-wall or registration
- Only general-purpose metrics (e.g., not e-commerce metrics only)
- Not a duplicate result already reviewed

6 Conclusions

In this chapter, we discussed metrics from the point of view of software startups. To give software startups ideas for what to measure, we compiled 118 metrics from literature (books, blogs, articles, etc.) written by various practitioner experts such as investors or startup incubator representatives. These metrics were divided into three categories: (1) software engineering metrics, (2) business and financial metrics, and (3) user and customer metrics.

Key takeaways

- ✓ Metrics help you make decisions based on data, not just intuition
- ✓ Metrics can and should be used together to gain a more in-depth understanding of your business.
- ✓ User and customer metrics are the most important metrics, although B2C and B2B startups should approach them differently.
- ✓ Software-as-a-Service services let you track your users as they use the service online. Use this to your advantage and gather data on their use habits in order to better understand them.
- ✓ Do not focus on vanity metrics such as total registered users: alone they do not tell you anything of value, although when combined with other metrics they can still provide value. A better metric than total registered users would be e.g. Monthly Active Users that actually shows you how many people still use the service.
- ✓ Differentiate between metrics for internal use and for stakeholder communication. Metrics that are very useful to your team may not be interesting at all to potential investors.
- ✓ Consider using method-specific (e.g., Scrum) or practice-specific Software Engineering metrics if you are using some common method(s) or practices.
- ✓ Gather data on possible applications of Big data analytics. Even if you do not have an immediate use for the data you are collecting, analyzing it in retrospect can help you discover surprising patterns.

Based on the data we gathered, we cannot make any sweeping claims as to what the best thing for your startup would be to measure. Different metrics are important for different software startups for different reasons. An early-stage software startup that has no users cannot measure user activity, and a B2B startup has far fewer customers than a B2C one. Taking this line of reasoning even further, metrics specific to your service may ultimately be the most important ones for your startup. If you are developing and operating an online game, you may be most interested in tracking how many of your users perform an action specific to your game every day after logging into it.

It can also be relevant to differentiate between metrics that are relevant to your startup team and metrics that are relevant from the point of view of your stakeholders. As software startups largely operate under very limited resources and are consequently reliant on outside funding especially early on, satisfying investors and potential investors is also important. While metrics related to tracking user behavior inside the software may be the most important ones for developing the software further, especially nontechnical investors are likely to be far more interested in metrics directly related to current revenue and future revenue forecasts.

References

1. McAfee, A., Brynjolfsson, E., Davenport, T.H., Patil, D.J., Barton, D.: Big data: the management revolution. Harv. Bus. Rev. **90**(10), 60–68 (2012)
2. Walker, J.S.: Big data: a revolution that will transform how we live, work, and think. Int. J. Advert. **33**(1), 181–183 (2014)
3. Unterkalmsteiner, M., Abrahamsson, P., Wang, X.F., Nguyen-Duc, A., Shah, S., Bajwa, S.S., Baltes, G.H., Conboy, K., Cullina, E., Dennehy, D., Edison, H., Fernandez-Sanchez, C., Garbajosa, J., Gorschek, T., Klotins, E., Hokkanen, L., Kon, F., Lunesu, I., Marchesi, M., Morgan, L., Oivo, M., Selig, C., Seppänen, P., Sweetman, R., Tyrväinen, P., Ungerer, C., Yagüe, A.: Software startups—a research agenda. e-Inform. Softw. Eng. J. **10**(1), 89–123 (2016)
4. Wang, X., Edison, H., Bajwa, S.S., Giardino, C., Abrahamsson, P.: Key challenges in software startups across life cycle stages. In: Sharp, H., Hall, T. (eds.) Agile Processes, in Software Engineering, and Extreme Programming. XP 2016. Lecture Notes in Business Information Processing, vol. 251. Springer, Cham (2016)
5. Croll, A., Yoskovitz, B.: Lean Analytics: Use Data to Build a Better Startup Faster. O'Reilly Media, Sebastopol, CA (2013)
6. Desjardins, J.: 34 Startup metrics for tech entrepreneurs. http://www.visualcapitalist.com/34-startup-metrics-founder-know/ (2017)
7. Gorski, T.: 21 Most important SaaS startup metrics. http://www.saasgenius.com/blog/21-most-important-saas-startup-metrics (2016)
8. Jordan, J., Hariharan, A., Chen F., Kasireddy, P. 16 Startup metrics. https://a16z.com/2015/08/21/16-metrics/
9. Straubel, E.: Getting funded: Part 5 (The metrics). Retrieved 07 Oct 2018 from https://www.bigroomstudios.com/startups/startup-metrics/
10. Ehrenberg, D. The seven startup metrics you must track. https://www.forbes.com/sites/theyec/2014/06/20/the-seven-startup-metrics-you-must-track/#2760f134725e (2014)
11. Lovelace, N.: How to measure your startup's success. https://medium.com/kandu/how-to-measure-your-startups-success-34b8aad7516b (2018)
12. AppsterHQ. 4 Financial metrics that all startups should measure. Retrieved 07 Oct 2018 from https://www.appsterhq.com/blog/4-financial-metrics-startups-measure/
13. Brookes, I.: Startup metrics for customer traction. https://www.cakesolutions.net/companyblogs/startup-metrics-for-customer-traction (2017)
14. Greenberg, O.: What are the best metrics to measure funded startup growth? https://kurve.co.uk/what-are-the-best-metrics-to-measure-funded-startup-growth/ (2016)
15. Kraus, E. The startup metrics cheat sheet: how to calculate what you are expected to know. https://www.mergelane.com/post/the-startup-metrics-cheat-sheet-how-to-calculate-what-you-are-expected-to-know (2016)
16. GrowthWright. Financial metrics every startup should measure. https://growthwright.com/blog/financial-metrics-every-startup-should-measure/ (2018)
17. Young Entrepreneur Council. 12 Success metrics your startup should be tracking. https://www.huffingtonpost.com/young-entrepreneur-council/12-success-metrics-your-s_b_3728052.html (2013)
18. Bloem, C.: 5 Performance metrics your small business should track. Retrieved 7 Oct 2018 from https://www.inc.com/craig-bloem/5-key-metrics-every-early-stage-business-must-track.html
19. Corporate Finance Institute. Startup valuation metrics (for internet companies). Retrieved 07 Oct 2018 from https://corporatefinanceinstitute.com/resources/knowledge/valuation/startup-valuation-metrics-internet/
20. Nadel, P. 12 KPIs you must know before pitching your startup. https://techcrunch.com/2017/02/04/12-kpis-you-must-know-before-pitching-your-startup/ (2016)
21. Parsons, N. What metrics to track (and what not to track). https://www.liveplan.com/blog/what-startup-metrics-should-i-track/ (2018)

22. Weiss, M.: Top startup traction metrics considered by seed round investors. https://www.rocketspace.com/tech-startups/top-startup-traction-metrics-considered-by-seed-round-investors (2017)
23. Cook, A.: Top 5 startup metrics to show traction. https://five23.io/blog/top-5-startup-metrics-to-show-traction/ (2018)
24. Singer, S.: How to measure your startup's performance (Pt. 2). https://magazine.startus.cc/measure-performance-startup-pt-2/ (2016)
25. Belosic, J.: All in the numbers: how to measure your start-up's success. https://www.themuse.com/advice/all-in-the-numbers-how-to-measure-your-startups-success (2018)
26. Tyson, L.: The ultimate startup metrics guide: 5 KPIs that VCs recommend. https://www.geckoboard.com/blog/ultimate-startup-metrics-guide-5-kpis-vcs-recommend/ (2016)
27. StartupBahrain. 7 Startup metrics you need to measure the growth of your startup in Bahrain. http://startupbahrain.com/newsfeatures/7-startup-metrics-need-measure-growth-startup-bahrain/ (2017)
28. Alexeeva, D.: 8 Startup metrics you should care about first. https://eze.tech/blog/8-startup-metrics-you-should-care-about-first/ (2018)
29. Causey, A.: How to measure your mobile app startup's performance. https://dzone.com/articles/how-to-measure-your-mobile-app-startups-performanc (2018)
30. Patel, N.: 9 Marketing metrics and KPIs every startup should be paying attention to. https://www.huffingtonpost.com/neil-patel/9-marketing-metrics-and-k_b_10769222.html (2016)
31. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Random House, New York (2011)
32. McNally, K., Odlum, N.: Finding the metrics that matter for your product. Retrieved 07 Oct 2018 from https://www.intercom.com/blog/finding-the-metrics-that-matter-for-your-product/
33. Middleton, M. 5 Key SaaS metrics every software startup should track. Retrieved 07 Oct 2018 from https://labs.openviewpartners.com/key-saas-metrics-to-track/
34. Harley, A.: Frequency & recency of site visits: 2 metrics for user engagement. https://www.nngroup.com/articles/frequency-recency/ (2016)
35. McClure, D.: Product marketing for pirates: AARRR! (aka Startup Metrics for Internet Marketing & Product Management). http://500hats.typepad.com/500blogs/2007/06/internet-market.html (2007)
36. Patel, N.: 9 Metrics to help you make wise decisions about your start-up. Retrieved 07 Oct 2018 from https://neilpatel.com/blog/9-metrics/
37. Dolginow, D.: Why product metabolism is every startup's first KPI. https://venturefizz.com/stories/boston/why-product-metabolism-every-startup-s-first-kpi (2011)
38. Almus, M., Nerlinger, E.A.: Growth of new technology-based firms: which factors matter? Small Bus. Econ. **13**(2), 141–154 (1999)
39. Kupiainen, E., Mäntylä, M.V., Itkonen, J.: Using metrics in agile and lean software development—a systematic literature review of industrial studies. Information and Software Technology. **62**, 143–163 (2015)
40. Warren, P., Gaskell, C., Boldyreff, C.: Preparing the ground for Website metrics research. In: Proceedings of the 3rd International Workshop on Web Site Evolution. WSE 2001 (2001)
41. van Moorsel, A.: Metrics for the internet age: quality of experience and quality of business. In: Proceedings of Fifth Performability Workshop, September 16, 2001, Erlangen, Germany. Hewlett Packard (2001)
42. Atterer, R., Wnuk, M., Schmidt, A.: Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In: WWW '06 Proceedings of the 15th International Conference on World Wide Web, pp. 203–212 (2006)
43. Ghanbari, H.: Investigating the causal mechanisms underlying the customization of software development methods. In: University of Jyväskylä: Jyväskylä Studies in Computing, vol. 258 (2017)

44. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: A systematic mapping study. Inform. Softw. Technol. **56**(10), 1200–1218 (2014)
45. Ross, S.A.: Uses, abuses, and alternatives to the net-present-value rule. Financial Management. **24**(3), 96–102 (1995)

Early-Stage Software Startups: Main Challenges and Possible Answers

Jorge Melegati and Fabio Kon

Abstract Software startups have a low probability of success. In their early-stage days, they face several challenges from different types that make it even harder to progress to the following stages. Some papers in the scientific literature focused on understanding these challenges. Meanwhile, others proposed solutions to these problems. In this chapter, we present a literature review of the challenges and patterns displayed in the scientific literature. Challenges are divided into four categories: related to product, market, finance, and team. The patterns presented in this chapter are (1) Get help from the methodologies, (2) Acquire customers, (3) Hack money incomes and outcomes, (4) Use available and simple tools, (5) Go up to the cloud, (6) Find your mentors, (7) Long-term purpose instead of money, and (8) Networking. We also show how the presented patterns could be used to tackle the identified challenges.

Keywords Software startups · Design patterns

1 Introduction

To create a software startup is a hard endeavor. Although there is no consensus about the failure rate, authors put this number as high as 75% or even 90%. Since the early days of the software startup phenomenon, there has been research interest in understanding why startups fail and what could be done to avoid it. For instance, Crowne [7] developed a model to tackle this problem dividing the evolution of

J. Melegati (✉)
Faculty of Computer Science, Free University of Bozen-Bolzano, Bolzano, Italy
e-mail: jmelegatigoncalves@unibz.it

F. Kon
Department of Computer Science, University of São Paulo, São Paulo, Brazil
e-mail: kon@ime.usp.br

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_8

a startup into three phases: startup, stabilization, and growth. Then, he identified common challenges for each stage and how it could be possible to tackle these challenges. More recently, Giardino et al. [13] performed a large-scale survey of 5389 responses and a multiple-case study to identify the challenges early-stage software startups face. In 2018, Berg et al. conducted a mapping study to organize software startups according to their characteristics [3].

Meanwhile, some initiatives proposed techniques to improve startups' practices. Some of these ideas were presented in the patterns community, a group of researchers that carries international conferences like PLoP (International Conference on Pattern Language of Programs) and their continental counterparts: EuroPLoP, AsianPLoP, and SugarLoafPLoP (in Latin America). In these conferences, researchers and practitioners get together to propose and improve reusable solutions to problems in software engineering and related areas through *writing workshops*. These solutions are presented in the form of a written pattern, describing a common procedure to tackle a set of similar problems. A set of related patterns focused on a specific context is called a *pattern language*.

The *pattern* term with this meaning was first used in the context of Architecture by Alexander [1] in their book *A Pattern Language: Towns, Buildings, Construction*, back in 1977 to present a set of solutions to recurrent architectural problems. This format was introduced in the software landscape almost 20 years later by Gamma et al. [12] with the seminal book *Design Patterns: Elements of Reusable Object-Oriented Software* where the authors presented standard ways to solve common programming problems and improve the quality of software architecture. Before this book, also known as GoF (the Gang-of-Four book), the computer science community did not have a common vocabulary to talk about high-level object-oriented constructions. Nowadays, most experienced developers know almost all of the GoF patterns and use their names as common words in their vocabulary, i.e., they talk among themselves in terms of Factories, Iterators, Composites, Observers, etc. The idea of patterns in software was so successful that it started to be applied to other areas of software engineering, such as architectural patterns, which led to a collection of several books on Pattern-Oriented Software Architecture.[1]

The idea of turning tacit knowledge into explicit knowledge in the form of patterns was applied to the way that teams develop software in the book *Organizational Patterns of Agile Software Development* by Coplien and Harrison [6]. Also, Mary Lynn Manns and Linda Rising used the pattern format to capture expert knowledge on how to introduce new ideas into organizations [18]. In these two cases, the patterns described social processes connecting people, usually, in a business environment.

Thus, the pattern format is a proven technique that could also be applied to the context of software startups. They are an easy way of making explicit, in an easy-to-understand way, the knowledge acquired by experts in the field in the past decades. In this century, tens of thousands of software startups failed while a few

[1] See http://www.dre.vanderbilt.edu/~schmidt/POSA.

thousand succeeded. In this process, the community learned a lot, but not always all the information is readily available to newcomers.

> Patterns are common procedures to tackle similar problems that occur in a specific context. In the case of early-stage software startups, some patterns mitigate challenges that threaten companies' success.

Researchers and practitioners in software startups have recently started to capture knowledge and write them in the form of patterns or pattern languages to improve their practices. Eloranta [11] made the first attempt to create a set of patterns to help startups to thrive. She described three patterns (Unique value proposition, Serve single customer segment first, and Develop only what is needed now) and envisaged seven more. Hokkanen and Leppanen [15] proposed three patterns to foster user involvement in startups. In the PLoP 2015 patterns conference, two different papers focused on software startups: Melegati and Goldman [19] proposed seven patterns, and Cukier and Kon [8] presented a draft of a pattern language. These authors came together to improve this nascent pattern language and published a follow-up paper in SugarLoafPLoP 2016 [10].

In this chapter, we present Cukier's patterns and how they can be used to solve or, at least, mitigate the challenges described by Giardino et al. [13]. Section 2 summarizes the challenges that early-stage software startups face and a classification based on their nature. Section 3 presents the patterns proposed and how they tackle the challenges presented in the previous section. Finally, Sect. 4 concludes the chapter.

2 Early-Stage Software Startups Challenges

Based on a large-scale survey with more than 5000 responses and 2 case studies, Giardino et al. [13] identified a series of challenges that early-stage software startups face. The authors divide them into four groups based on MacMillan et al.'s four classes model of new ventures failures [17]. The challenges are related to the product, market, team, and finance. Table 1 presents a summary of them.

Table 1 Summary of challenges that early-stage software startups face

Category	Challenge
Product-related challenges	Thriving in technology uncertainty
	Time spent developing features that users are not interested in
Market-related challenges	Did not define and validate the need of potential customers
	Attract customers
Team-related challenges	Lots of activities in a short time
	Build entrepreneurial team
Financial challenges	Keep the business running to break-even point

Product-Related Challenges This category encompasses difficulties related to the creation and evolution of the product or service offered by the startup to their users or customers. Problems put in this category are linked to startups' innovative nature that represents a strength but also a weakness of these new companies. The newness to the team of technologies used to develop the product is probably the toughest challenge for these companies, what Giardiano et al. called *thriving in technology uncertainty*. Crowne [7] had already mentioned this problem calling it "product platform" consisting of "hardware and software components that underpin the product" being several times "not understood, discussed, and managed."

Another challenge in this category is related to the waste of time developing features that the users do not want [13]. Ries [22] tells the story of his startup IMVU that inspired him to write his commercially successful book where he presented the Lean Startup methodology. At the time the company was created, there were several instant messengers (IM) networks available, and the startup founding team thought that having different applications to connect to each network annoyed users. Based on this belief, they developed a product that would allow users to use several networks in a single application. When they observed that nobody was using it, they performed interviews with several users to understand why. They realized that this feature was not useful to the users. Instead, they used several IM applications to organize their friends in different groups. Nguyen-Duc classified barriers to prototyping speed into five groups (1) artifacts, (2) team competence, (3) collaboration, (4) customer and (5) process dimensions [20].

Market-Related Challenges In this category, problems concern reaching their potential market, through exploiting the opportunities and getting a bigger number of customers. The first challenge is close to the last one in the previous category. Giardino et al. observed that software startups "do not define and validate the needs of their potential customers." That is, most of the times, founders create their companies to build a product idea, but they had not identified a market (customer) need. Developing features that nobody wants and not understanding the potential customers' needs are two sides of the same coin. That is, instead of looking for the customers' real needs and developing a product to fulfill this need, startups often focus on their initial non-validated idea and spend their scarce resources building a product that probably nobody will pay for. Bajwa et al. described several cases that startups need to change courses of action due to market concerns [2]. Figure 1 presents a summary of the differences and similarities of these problems.

Not validating an idea is also related to another problem in this category: to attract customers. Without customers, especially paying ones, the company does not generate revenue, and it cannot even attract investors to support the further development of the company. The startup is doomed to fail. Besides that, the team is lost, not able to tell if the product or the chosen distribution channels are the right ones.

Early-stage startups face challenges from four types: related to the product, to the market, to the financial aspect, and to the team.

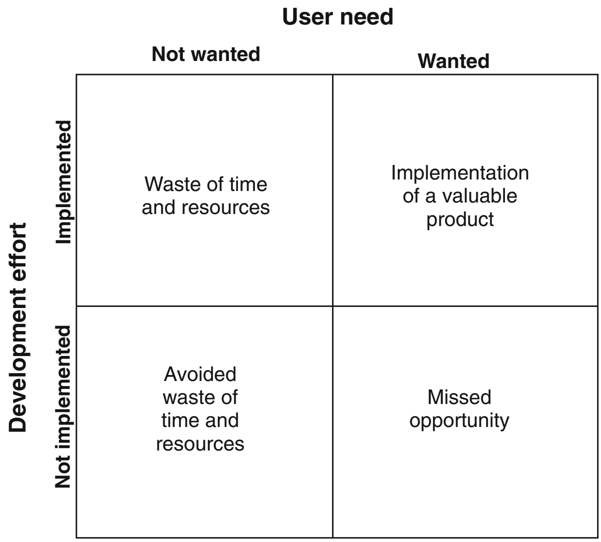

Fig. 1 Challenges in product and market areas related to the product creation and evolution

Financial Challenges A well-known characteristic of an early-stage startup is the lack of financial resources. Especially before getting an investment, teams had to persevere with their own money or from family and friends. They have to set up a minimal structure, pay the founders or early employees a salary without the income a conventional company has. In this category, the identified challenge is to keep the business running to break-even point, that is, when revenue surpasses costs.

Team-Related Challenges Finally, the last category concerns the team composition, capabilities, and the tasks that are presented to it. The two main problems are the significant number of different activities the team has to perform in a short time and the difficulties in bringing more people onboard. The first is a consequence of setting up a new company and all the duties that it brings constrained by a tight budget. For instance, legal and accounting tasks should be performed, but the startup may not have resources to hire specialized people or outsource it to another company. Most of the times, the founders have to handle it themselves. Nevertheless, the team still needs a good pool of talent to build the company core: software, and also other related competencies like user interface design and marketing. However, in this market, startups have to compete with consolidated companies that can offer more money and stability for a small pool of talented people.

3 Patterns

In this section, we will present patterns proposed by Cukier et al. and how they tackle the challenges described in the previous section. The classical form of presenting patterns is called the *alexandrian* way in reference to Alexander et al. book. This format is primarily composed of: context, problem, forces, solution, consequences, and known uses. There are other ways to represent patterns like, for instance, stories. In this chapter, patterns are presented in an adapted *alexandrian* format. Since the context of a software startup and forces acting on them have already been extensively discussed in this book, they will not be described for each pattern. Instead, the pattern description starts with the solution followed by which problems presented in the previous section it mitigates. Then, the consequences of the pattern use are presented, both positive and negative. Finally, a small list of known applications is presented. Figure 2 shows a summary of patterns and which challenges they tackle.

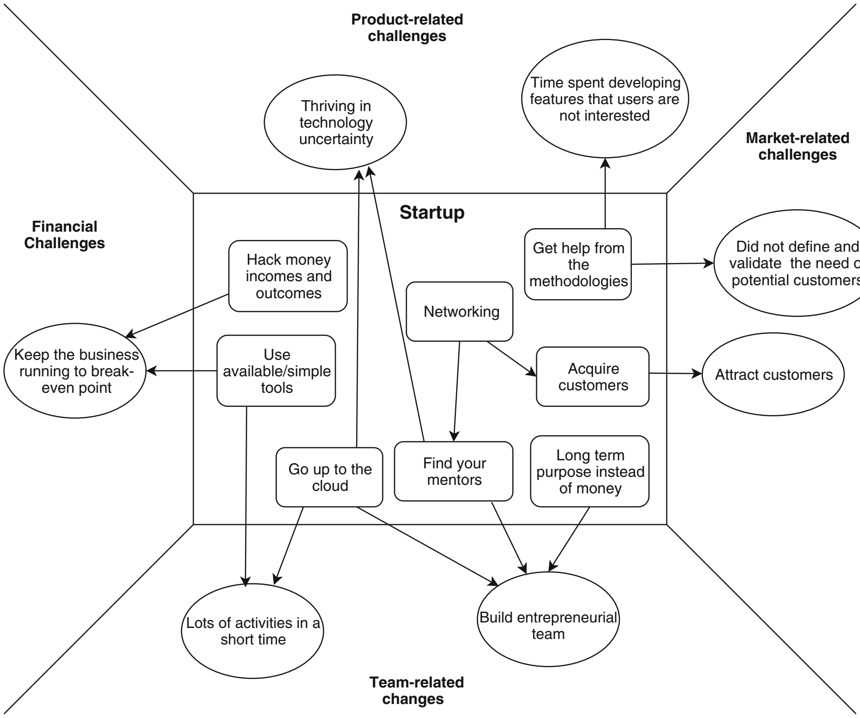

Fig. 2 Challenges (in oval) for early-stage software startups and patterns (in boxes). Arrows represent which challenges a pattern tackles

3.1 Get Help from the Methodologies

Use Startup Development Methodologies to Avoid Mistakes that Several Other People Already Did Several books propose methodologies to support startups creation and to help them find their way to success. Lean Startup [22] and the series of other books based on it are an example. Customer Development [4], a precursor of Lean Startup, is another example. In summary, these methodologies focus on not implementing all the software upfront but, instead, listen to your customers and try to understand their needs and if they want your product. Startups still do not use it as often as shown by Pantiuchina et al. [21] but, instead, often spend a vast amount of time developing the solution instead of understanding their customers [14].

In the specific context of software startups, Agile methodologies have been seen as useful approaches. They embrace change instead of avoiding what is common in such an innovative environment. Although these methodologies are increasingly popular, founders with previous experiences on traditional software methodologies could be tempted to follow what they already know.

Besides the previously mentioned Lean Startup and Customer Development, other methodologies focused on designing new products might be useful. Design Thinking [5] is a set of techniques used by designers for product development like, for instance, ideation. Design sprint [16] is a 5-day process to answer crucial business questions through prototyping and customer feedback. While using these techniques, Agile methodologies, e.g., XP and Scrum, should be useful to guide software development since they embrace change instead of avoiding it. To describe these methodologies is beyond the scope of this chapter, but an interested founder or early employee could look for more information about them. Teams might choose to use a set of techniques from different methodologies instead of strictly just following one. For instance, they may follow a Lean Startup Build–Measure–Learn cycle using Design Thinking ideation to generate hypotheses and setting up Scrum sprints.

Problems Tackled These methodologies offer tools to deal with the two related problems: *time spent developing features that users are not interested (product)* and *did not define and validate the need of potential customers (market)*. These problems are especially crucial in early-stage startups that are still looking for a sustainable business model and struggle with constrained resources. Nevertheless, more mature startups could use them to, for example, create new features or test new markets. Using these tools, startups will better understand their customers and build what is important for them, saving resources, and better exploring the market opportunities.

Consequences As a positive consequence, software startups using these methodologies have a better chance of succeeding or, at least, failing faster and saving founders' time and resources. On the other hand, if they do not correctly implement the practices, wrong results could lead the team to a wrong path. For instance, not well-planned experiments could give an opposite conclusion from the reality, and founders are prone to trust this result.

Known Uses The books presenting these methodologies display several cases. Ries' IMVU and Blank's consultancy cases are examples. Besides that, there are several books about these methodologies like those in *The Lean Series*.

3.2 Acquire Customers

Know Your Customer and Market and Use Multiple and Adequate Channels to Reach Them There are several possible channels to reach customers. A startup should have a comprehensive set of channels to market its product according to its stage, market, and budget. It is not possible to describe all the possibilities in this chapter; we focus on some of them. First, the company should monitor its position on search engine results. A buzzword in this field is SEO (Search Engine Optimization). It encompasses a set of techniques to improve a website result rank in search engines or mobile app stores. Similar techniques could be used to improve results on other so-called organic (not-paid) channels like social media. If the company already has resources to spend on marketing, several other channels arise. For instance, startups could pay for ads in search engine results or blogs related to their products. In this regard, some practices allow the company to wisely spend money on ads, e.g., the so-called SEM (Search Engine Marketing) techniques. A fundamental concept in this field is the ROI (Return on Investment), that is, how much money the company makes from a set of customers versus how much it spends to bring those customers. This metric could be used to compare channels. Affiliate programs where the company pays a commission based on leads or sales could also be an option. Regarding branding, startups may work with a press office to have their product on TV news, newspapers, or news sites.

Problems Tackled The startup should choose among these techniques those that best-suit its market and business model, addressing the problem of *acquiring customers (market)*. Nevertheless, the team should pay attention to really create value to the users and not just paying for bringing customers without engaging them with the product. Although it may change throughout its history, a startup should have a marketing plan: in the beginning to bring users to test their product and later to acquire new customers and engage the current ones.

Consequences The use of different marketing channels will bring more customers, but it will demand the acquisition of knowledge and other abilities. This requisite could increase the number of tasks and the need for different people in the company. Besides that, it could also represent more money to be spent. Another concern is that the number of users may not be the correct metric to observe. For instance, if you pay enough money, you will bring users to your website but not necessarily they are engaging with the product. Therefore, the startup will not be validating their idea. The number of users is an example of what Ries called a vanish metric.

Known Uses As in the previous patterns, several books present techniques and cases of success. For instance, there are several books about SEO techniques and how it is possible to get many users with low investment, by just changing your website. Another area full of examples is digital marketing.

3.3 Hack Money Incomes and Outcomes

Look for Creative Ways to Save Money in Noncore Needs There are several creative ways to save money. These solutions represent a relevant strategy to save money for startups that face a lack of resources. Founders have to look for ways to not spend money on noncore activities. Whenever a big spend has to be made, the founders should consider cheaper options. More expensive options consume money that could be used elsewhere like the product development reducing the time the company has the capital for running. In particular, recurrent expenses, i.e., those paid monthly or yearly, may pose a more significant threat since they represent a constant budget pressure. Cukier et al. give several examples: a big office if people can start working from home or using a free co-working space; buy new and premium equipment and furniture if second-hand are available.

Problems Tackled Saving money will help the startup to *keep the business running to break-even point (financial)*.

Consequences Through these hacks, software startups can save money to spend in their core: product development. Nevertheless, the lack of a proper structure could hinder trust in the company: future employees, investors, or customers could not get a good impression from a company if it is not well presented.

Known Uses Cukier et al. [10] mentioned the case where one of the authors bought used furniture for his company's office to save money. They also mentioned programs targeted to startups by cloud providers that give an amount of money to be spent in the platform for a while.

3.4 Use Available Tools

Look for and Use Simple, Online or Offline, Tools to Help You Do Your Work, Especially Those Noncore Activities Creating a company will require, sooner or later, to incorporate a formal entity. At this moment, the founders will have several new tasks like accountancy and finance. This requisite represents an increased workload on founders representing less time that they can invest in the product itself. Startups should take advantage of several online, or even offline, tools to support noncore tasks. For instance, they can use an online platform to handle the accountancy and emit invoices. The use of such tools will also represent an economy of hiring people to do these tasks.

Problems Tackled The use of tools to perform noncore tasks will decrease the number of jobs the team should do, diminishing the problem of *lots of activities in a short time (team)*. Besides, the money saved using these tools instead of hiring more people will make it easier to *keep the business running to break-even point (financial)*.

Consequences Using available tools, software startups will probably have their product running faster and cheaper. Nevertheless, these tools may not be robust and may have to be replaced during the company's lifetime. In this case, part of the cost is transferred to a later time for the company. Sometimes, the transition to a different tool may be complicated and hard to achieve.

Known Uses Nowadays, plenty of tools are available online to support companies, and startups could take advantage of them. Examples are SaaS tools for accounting or even recruiting.

3.5 Go Up to the Cloud

Whenever Possible, Use Cloud Solutions to Provide Noncore Features of Your Business In the last years, there is a clear trend toward IT products as services. These solutions include not only software as the Software-as-a-Service (SaaS) but also Platform-as-a-Service (PaaS) and even Infrastructure-as-a-Service (IaaS). For instance, in the latter, you can get a server in a hosting company available to get your solution deployed with few clicks. These solutions represent a huge advantage for startups since they do not need to invest resources, for instance, to buy servers, and avoid hiring more people in the early stage to handle tasks related to colocated servers.

Different solutions provide different levels of control and maintenance time. SaaS solutions, for instance, a Content Management System (CMS), does not take time to be used but does not enable too much customization. These tools are useful for early-stage startups to test an idea or a prototype. As the product evolves, it becomes hard to adapt the tool to the needs. In a next level, PaaS platforms, solutions that allow to upload the solution code but do not give access to the underlying infrastructure, give the team more freedom regarding features without the burden of managing the infrastructure. Finally, IaaS platforms provide a higher level of control but pose more maintenance-related tasks. A startup team should compare the numerous available solutions and choose the one that is most suitable considering these aspects and considering a plan to perform once the user base and the number of expected features grow.

Besides that, several SaaS solutions exist for a big range of tasks, from communications infrastructure like e-mail to team and project management tools to source code version systems. Selecting tools that provide these features saves money and human resources from startups that do not have to handle them in-house.

Problems Tackled Through outsourcing several tasks like infrastructure and other noncore software, the startup team mitigates the problems of *lots of activities in a short time (team)* and *build entrepreneurial team (team)*.

Consequences Avoiding the development of some pieces and the setup of computing infrastructure will save money and time for the software startup. Nevertheless, this will make the solution be tailored to the selected cloud platform, which increases the risk of lock-in. Another threat is the increasing costs when the product starts to be more used: given that these platforms generally use pay-per-use mechanisms, the infrastructure cost could rise a lot.

Known Uses There are several cloud solutions providers, and most of them offer some incentive for startups for a limited time. These offers could be useful to test an idea, spending virtually nothing with hardware infrastructure.

3.6 Find Your Mentors

Search for and Connect with Experienced People That Could Help You to Build Your Company The number of different capabilities needed by a software startup is large. Seppanen et al. [24] classified them as application domain, software development, hardware development, mechanics development, systematic development work, and difficult technology domain. Nevertheless, founders are generally focused on the innovation [23] what, in the case of software startups, is software. There are several other tasks not related to the main competencies of the founders, and usually, the amount of work is not enough to hire a person. Besides that, startups still struggle with lack of resources. Experienced people could be asked to work as advisers or mentors to the team in different areas: technology, business, marketing, legal, etc. Mentors are common in accelerator and incubation programs where these experienced professionals and entrepreneurs advise on a wide range of topics from legal aspects to user experience tips. Although not all startups are able to participate in an accelerator program, they could look for advisers through their contact networks or in entrepreneurship events. Several mature entrepreneurs are willing to share their experiences with others. If needed, founders could offer a small share in the company to give an incentive for the mentor to participate in the startup life actively.

Problems Tackled The help of externals as mentors could be a mitigation strategy to *build entrepreneurship team (team)* since their expertise could be used as itself or through training the team instead of hiring other people from outside.

Consequences Through mentors' knowledge, the expertise range of the team gets broader without huge investments and hiring. Nevertheless, the commitment may be lower than a full-time employee or founder. Besides that, a mentor could drop out of the project, and the team will face that expertise loss. Therefore, it is essential for the team members to begin absorbing part of the knowledge and internalize it in the company.

Known Uses The concept of mentors and advisers is common in acceleration programs and incubators, where experts in different areas are invited to talk to companies and help them. Several times these mentors could become partners of the companies.

3.7 Long-Term Purpose Instead of Money

Offer Equity, Freedom, and Job with Purpose Instead of High Salaries Software startups have to compete with several other consolidated companies for a small pool of talent, especially the technical one. This challenge is even a more significant problem given the lack of resources they face. Then, startups should differentiate from consolidated companies offering other incentives. For instance, Seppanen et al. [24] mentioned "in [three cases], the missing economic resources led to offering shared ownership instead of normal salaries when hiring new team members." The most common type of incentive is equity, that is, shares of the company. From an economic perspective, these shares could represent a possible financial outcome in the future if the company pays dividends, is sold, or becomes public. In such a way, employees are encouraged to work hard for the company's success, so they will also have a financial outcome. Besides that, there is a psychological effect of ownership, that is, working on something that is also theirs. Another way of stimulating team members is through a less constrained and more joyful working environment. Some examples of implementing it are flexible working hours, free food on the office, for example, breakfast, and social events.

Problems Tackled This pattern is especially useful for the *build entrepreneurial team (team)* challenge.

Consequences Through a better proposal consisted of equity and quality of life incentives, startups can build a better team with a smaller initial investment. On the other hand, equity distribution could dilute the founders' participation and could represent a lower profit in an exit strategy. A too informal environment could reduce productivity and could hinder a response in a stressful period.

Known Uses Equity distribution is a popular method to compensate for the participation of a co-founder or to reduce an early employee in a startup.

3.8 Networking

Build a Strong Contact Network Among Professional in Several Areas Related to Your Business and the Startup Ecosystem Meeting people is extremely important for startup founders. Other people's experience could represent valuable learning, and new contacts could turn into sales or investments in the company. To participate in events organized within the startup ecosystem [9] where the startup operates or even in other places could create these connections needed. Several institutions or groups of people organize network events. Some universities promote events to foster entrepreneurship within their students and the surrounding environment. These events could also be useful to find students to do an internship or recent graduates to work in the company. Co-working spaces may also organize gatherings between their users. Other governmental or nongovernmental institutions may promote events to strengthen the ecosystem. Founders should be prepared to pitch or present their ideas to potential customers or collaborators. There are several tips online on how to do it for different durations.

Problems Tackled Although this pattern does not directly address any of the identified challenges, it could be useful to help to implement other patterns like *find your mentors* and *acquire customers*.

Consequences Through contact networks, founders can have access to several people with different expertise and resources. These contacts allow the team to implement the previous patterns better. However, participation in events takes time and energy that the founder is not directly using to develop the product. There should be a balance to not hinder the development.

Known Uses There are several startup events from small meetups to big conferences that connect entrepreneurs, investors, aspiring employees, accelerators, and government agents. In these events, founders can meet possible co-founders, funding, or other partnerships.

4 Conclusions

This chapter presented the challenges that early-stage software startups face and a set of solutions in the form of patterns on how to tackle these challenges. These collections of challenges and solutions are not final, and different problems may arise for companies depending on their specific context, like market and geographical ecosystem [9].

Regarding the limitations of this approach, there are two major issues that must be considered: (1) it is not always obvious which patterns to use and whether that pattern is valid in the specific context of each startup and (2) there is no real guarantee that the patterns found in the literature are really proved solutions that can be extensively applied with successful results. To tackle the first issue, entrepreneurs must try to educate themselves the best they can and also use the knowledge of more experienced people like mentors and investors. The danger is that the solutions proposed may not work well in some contexts. For instance, a startup developing a product for a real-time environment like avionics has to follow a series of strict guidelines that are not compatible with several solutions presented here. Nevertheless, these lists present challenges that the majority of software startups face in their early days as researchers observed and a set of possible solutions to these problems.

The patterns community typically mitigates the second issue by submitting patterns and patterns languages to a series of checks over months or even years by both experts in the domain and experts in pattern writing. The patterns described in this chapter, for example, were written based on years of experience of software developers and academics in the field of software startups, and they were revised over several writing workshops.

However, the field of patterns for startups is still in its infancy, and the existing patterns are far from the maturity reached by other pattern communities. We expect that practitioners and researchers will contribute to the area in the next years, producing more and better patterns. In this way, in the future, young entrepreneurs would have a more natural way of acquiring practical guideline to apply to their daily, challenging lives.

Key findings

Patterns for early-stage startups present solutions to common challenges these teams face. To implement these solutions, a practitioner should look for different solutions, search for more information about various tools that implement the solutions, and choose the one that best suits their needs.

References

1. Alexander, C.: A Pattern Language: Towns, Buildings, Construction. Oxford University Press, Oxford (1977)
2. Bajwa, S.S., Wang, X., Duc, A.N., Chanin, R.M., Prikladnicki, R., Pompermaier, L.B., Abrahamsson, P.: Start-ups must be ready to pivot. IEEE Softw. **34**(3), 18–22 (2017). https://doi.org/10.1109/MS.2017.84
3. Berg, V., Birkeland, J., Nguyen-Duc, A., Pappas, I.O., Jaccheri, L.: Software startup engineering: a systematic mapping study. J. Syst. Softw. **144**, 255–274 (2018). https://doi.org/10.1016/j.jss.2018.06.043. http://www.sciencedirect.com/science/article/pii/S0164121218301286

4. Blank, S.: The Four Steps to the Epiphany: Successful Strategies for Products that Win. BookBaby (2013)
5. Brown, T.: Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation. HarperCollins e-books, New York (2009)
6. Coplien, J.O., Harrison, N.: Organizational Patterns of Agile Software Development. Pearson Prentice Hall, Upper Saddle River (2005)
7. Crowne, M.: Why software product startups fail and what to do about it. Evolution of software product development in startup companies. In: IEEE International Engineering Management Conference, vol. 1, pp. 338–343 (2002). https://doi.org/10.1109/IEMC.2002.1038454
8. Cukier, D., Kon, F.: Early-stage software startup patterns strategies to building high-tech software companies from scratch. In: Pattern Language of Patterns, pp. 1–11 (2015)
9. Cukier, D., Kon, F.: A maturity model for software startup ecosystems. J. Innov. Entrep. **7**(1), 14 (2018)
10. Cukier, D., Kon, F., Melegati, J.: Early-stage software startup patterns—towards a pattern language. In: 11th Latin American Conference on Pattern Languages of Programs (SugarLoaf-PLoP'16), pp. 1–16. Buenos Aires, Argentina (2016)
11. Eloranta, V.P.: Towards a pattern language for software start-ups. In: 19th European Conference on Pattern Languages of Programs, pp. 1–11 (2015). https://doi.org/10.1145/2721956.2721965
12. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design patterns: elements of reusable object-oriented software. In: Addison-Wesley Professional Computing Series. Pearson Education, Milano (1994)
13. Giardino, C., Bajwa, S.S., Wang, X., Abrahamsson, P.: Key challenges in early-stage software startups. In: Lecture Notes in Business Information Processing, vol. 212, pp. 52–63 (2015). https://doi.org/10.1007/978-3-319-18612-2_5
14. Gutbrod, M., Münch, J., Tichy, M.: How do software startups approach experimentation? Empirical results from a qualitative interview study. In: Lecture Notes in Computer Science, vol. 10611, pp. 297–304. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69926-4_21
15. Hokkanen, L., Leppänen, M.: Three patterns for user involvement in startups. In: Proceedings of the 20th European Conference on Pattern Languages of Programs, pp. 1–8. ACM, New York (2016). https://doi.org/10.1145/2855321.2855373
16. Knapp, J., Zeratsky, J., Kowitz, B.: Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days. Simon & Schuster, New York (2016)
17. MacMillan, I.C., Zemann, L., Subbanarasimha, P.: Criteria distinguishing successful from unsuccessful ventures in the venture screening process. J. Bus. Ventur. **2**(2), 123–137 (1987)
18. Manns, M.L., Rising, L.: Fearless Change: Patterns for Introducing New Ideas. Addison-Wesley, Boston (2005)
19. Melegati, J., Goldman, A.: Seven patterns for software startups. In: Proceedings of the 22nd Conference on Pattern Languages of Programs (2015)
20. Nguyen-Duc, A., Wang, X., Abrahamsson, P.: What influences the speed of prototyping? an empirical investigation of twenty software startups. In: Baumeister, H., Lichter, H., Riebisch, M. (eds.) Agile Processes in Software Engineering and Extreme Programming. Lecture Notes in Business Information Processing, pp. 20–36. Springer, Cham (2017)
21. Pantiuchina, J., Mondini, M., Khanna, D., Wang, X., Abrahamsson, P.: Are Software Startups Applying Agile Practices? The State of the Practice from a Large Survey, vol. 283, pp. 167–183 (2017). https://doi.org/10.1007/978-3-319-57633-6_11
22. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business, New York (2011)
23. Seppänen, P., Oivo, M., Liukkunen, K.: The initial team of a software startup. In: 2016 International Conference on Engineering, Technology and Innovation (ICE)\IEEE International Technology Management Conference, pp. 57–65 (2016)
24. Seppänen, P., Liukkunen, K., Oivo, M.: Little Big Team: Acquiring Human Capital in Software Startups, vol. 10611, pp. 280–296 (2017). https://doi.org/10.1007/978-3-319-69926-4_20

Part III
Startup Ecosystems

The Roles of Incubators, Accelerators, Co-working Spaces, Mentors, and Events in the Startup Development Process

Nirnaya Tripathi and Markku Oivo

Abstract This chapter aims to explore supporting factors, such as incubators, accelerators, co-working spaces, mentors, and events in the startup ecosystem. To understand these five aspects and to explore their roles in startups, we investigated an Oulu startup ecosystem. In this case study, we conducted research interviews with practitioners working with startups, accelerators, incubators, venture capital firms, and co-working space organizations. By using real case examples, the results discussed in this chapter can help entrepreneurs understand the commonalities and differences between incubators and accelerators, their types (university-based or profit/nonprofit), the kinds of business ideas, and the entrepreneurs they focus on. Furthermore, the roles of co-working spaces, mentors, and events and their effects on entrepreneurs and startups are discussed. At the end of the chapter, we also show the interrelationships between these five aspects in the Oulu startup ecosystem and their influence on different startup development stages.

Keywords Incubator · Accelerator · Co-working space · Mentor · Events · Startup ecosystem · Startup

1 Introduction

The financial crisis and the economic recession at the end of the last decade led to high unemployment figures, forcing researchers and governments to focus their attention on the process of job creation [1, 6]. Scientific studies, such as [1, 2, 9], have found evidence that new businesses such as startups create the majority of the net jobs compared to established companies. However, it has been found that 90% of early-stage startups fail during their first 2 years due to various reasons, such as a lack of problem-solution fit and a failure to learn from mistakes and face challenges,

N. Tripathi (✉) · M. Oivo
M3S Research Unit, University of Oulu, Oulu, Finland
e-mail: nirnaya.tripathi@oulu.fi; markku.oivo@oulu.fi

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_9

Table 1 A brief description of supporting factors

Supporting factors	Role
Incubator	To run programs (1–2 years duration) to assist early-stage startups [11]
Accelerator	To run short, intensive programs (3–6 months duration) to assist early-stage startups [11]
Co-working space	To provide places to work and collaborate with other startups [11]
Mentor	To guide and coach startup founders and team members to achieve the necessary skills for the business and product development [11]
Event	Activities happening at specific times and in specific places to enable collaboration and knowledge sharing

such as building the product, finding team members, developing a business model, creating a minimum viable product, etc. [7, 13, 18].

Thus, in order to support startups and entrepreneurs with these challenges and avoid failures, a suitable ecosystem needs to be built to support early-stage startups and to highlight a region's speciality in terms of innovation and startup [8, 17]. Through a startup ecosystem, a successful startup can be created in a region. Our earlier work [17] conducted a multivocal literature review on the topic, in which various regional and national cases of startup ecosystems were examined and analyzed. One conclusion from the study was that the startup ecosystem is a regional phenomenon, and sub-elements (see Table 1 for description), such as incubators, accelerators, co-working spaces, mentors, and events, act as supporting factors; thus, they can play crucial roles during the early-stage startup's development [11, 17].

However, a clear understanding regarding the roles and interrelationships between these five sub-elements through a case study is currently lacking. For example, previous studies, such as [3, 4, 10, 14], have explored these aspects separately and not in conjunction through a suitable startup ecosystem case. To address this, our chapter discusses these aspects by taking a startup ecosystem in Oulu city[1] as a research case (see Sect. 3.2). In Oulu, every year hundreds of startups are created, in which many of those startups get assistance from supporting factors during their early stages. These supporting factors such as incubators (see Sect. 2.1), accelerators (Sect. 2.2), co-working spaces (Sect. 2.3), mentors (Sect. 2.4), and events (Sect. 2.5) are examined and analyzed for their roles in the startup ecosystem and their influence on early-stage startups.

The chapter is divided into four main sections. In Sect. 2, we provide descriptions of these five sub-elements along with examples, as observed in the Oulu startup ecosystem. In Sect. 3, we briefly summarize related work and our research method process to support our claims. Finally, in Sect. 4, we discuss the interrelationships between these sub-elements and the study's implications for entrepreneurs and researchers.

[1] https://www.businessoulu.com/en/frontpage/en/company-networks-2/businessoulu-startup.html.

2 Supporting Factors in the Oulu Startup Ecosystem

2.1 Incubators

To become an entrepreneur and build a startup, a person (experienced or inexperienced) must invest effort and time in order to learn entrepreneurship skills, find/validate different business ideas by discussing them with other people, and identify scalable business ideas. The process could also result in finding potential founding team members and the development of a prototype or a minimum viable product. [C05] The word "Incubator" means to incubate; thus, their main aims are to nurture young entrepreneurs and to raise business ideas for early-stage startups [4]. In the literature, incubators are also described as programs of 1–2 years duration [11] and are classified into three types [4]: *university-based incubators*, *profit*, and *nonprofit* (e.g., government, community-based incubators). They provide services such as *infrastructure*, *coaching*, and *networking* [14].

University-Based Incubators Two higher education institutions, the University of Oulu (UoO)[2] and Oulu University of Applied Science (OUAS),[3] are actively promoting entrepreneurship and startup culture and mindset among their students. For example, OAMK LABS[4] is a pre-incubator at OUAS, offering incubator programs to participants to develop new skill sets and create new business and innovative solutions [C05]. The business idea can be general, and the participants get a mentor to develop the business idea further. In addition, Kickstart Oulu conducts a business idea competition,[5] which enables participants to demonstrate their business ideas and to be evaluated by external expert judges. Showing their business ideas can encourage participants to advance their business ideas further into potential startups [C02]. At the University of Oulu, the Business Kitchen[6] incubator works on incubating the students' business ideas. Furthermore, networking services provided by incubators, such as the Business Kitchen, enable early-stage startups to meet appropriate contacts, which aids in the development of some product components, since startups' limited resources may not allow them to build the whole product [C10]. Moreover, university-based incubators can also assist with the transferral of intellectual assets from university personnel to startups that are intending to commercialize those intellectual assets [4].

[2] https://www.oulu.fi/university/innovations-and-entrepreneurship.

[3] https://www.oamk.fi/en/studies-and-applying/masters-degree/education-entrepreneurship/.

[4] OAMK LABS: https:/oamklabs.fi.

[5] https://www.kickstartoulu.fi/business-idea-competition.

[6] Business kitchen: https://www.businesskitchen.fi/.

Nonprofit/Profit-Based Incubators For experienced professionals, organizations such as Starttaamo[7] and Yritystakomo[8] exist, in which participants discuss their business ideas with other members whom they trust. Thus, they receive feedback on their ideas to determine which business idea is worth developing further [C13].

Another incubator is Kielo Growth,[9] which is fee-based and offers services such as infrastructure and mentorship. Kielo creates an environment in which startups (acting as tenants) help each other by discussing the product and the business development. This, in turn, leads some startups to become partners to offer more value to customers [C13-C15]. Furthermore, incubators' managers can assist startup companies in networking with investors, legal personnel, and accounting advisors, and technology transfer [4].

Another interesting observation was that some experienced startup founders, such as [C01], [C06], [C08], did not seek assistance from an incubator and only used their infrastructure to fulfill their operations. In addition, some reasons why an incubator could fail (as discussed in [14]) include:

– Incubators mostly provide infrastructure services rather than coaching and networking; and
– Alternative resource preferences for entrepreneurs and startup companies are limited.

2.2 Accelerators

The objective of an accelerator is to speed up the startup development process. Thus, the main difference between an accelerator and an incubator is the accelerator's short intensive programs, which are usually between 3 and 6 months duration [4]. Accelerators can function by experienced business people [15] assisting in mentorship, creating the startup team members, and providing seed money. They can also offer a co-working space to develop their primary product, and they can help with recognizing potential customers for the product idea, networking with key people, providing instruction on search engine optimization, and creating events to pitch the business ideas to investors [4, 15].

The two accelerators in the Oulu startup ecosystem were Avanto[10] and Nestholma.[11] Once the potential scalable business idea is identified during the incubator's program, startup founders can take the assistance of these two accelerators to develop the business idea further. The two accelerators have similar work processes and connections with other stakeholders in the startup ecosystem; however, a key difference is that Avanto focuses on the early stage of the business idea while Nestholma focuses on the advanced stage [C02].

[7] Starttaamo: http://www.starttaamo.fi/.

[8] Yritystakomo: http://www.yritystakomo.fi/.

[9] Kielo growth: https://kielo.com/kielo-in-english/.

[10] Avanto: https://www.oulu.fi/forstudents/entrepreneurship/avanto-accelerator.

[11] Nestholma: https://nestholma.com/collaboration-programs/oulu-startup-accelerator/.

Avanto is a university-based accelerator that conducts programs of 1–3 months duration and mainly focuses on business ideas that are scalable and have the potential to make money fast [C02]. During the program, different tools, such as the Lean Startup methodology, design thinking, and service design thinking, are used to shape the business idea and to assist with business idea validation and business model development through the help of experts [C04,C11,C18]. Participants are also provided with a co-working space to build a team [C02,C11]. Avanto cooperates with incubators such as Business Kitchen and successful local startups to invite them as mentors during the program. Furthermore, university researchers, who are equipped with technical expertise but who lack business-related knowledge, can get assistance through Avanto programs to create spin-off startups. At the end of an accelerator program, a big event "demo day" is conducted where participants need to do a pitch in front of a large audience and investors [4]. A similar event was also observed in the Avanto program, which ended with a big event. After the program, participants receive study credits for their participation in the program [C12], and more connections and information are provided, which could lead to further development [C10].

The other accelerator, Nestholma, focuses on experienced persons and provides services similar to Avanto [C03]. For some startup founders, they assist with providing the necessary information during the early stages for further development [C17]. Furthermore, some experienced founding members of startups [C01,C06,C08] did not need the help of accelerators in the startup ecosystem since they were self-sustainable and could support themselves. As interviewee C01 stated:

We have been self-sustainable that way that we didn't use any accelerator, and the simple reason was that our founder was more experienced than plenty of those people who were working in those accelerators. He knew the things already better than those guys.

Concerning the programs, Cohen and Hochberg [4] mentioned that accelerator programs can focus on the usual startups, but they can also target some specific types of startups, for example, energy startups, education-related startups, or healthcare-related startups. In addition, Haines in [8] discussed the following types of programs to promote startups and innovation in the region: *school-based (10-week duration)*, *Startup Basecamp (8-week duration)*, and *programs to connect startups with small- and medium-sized enterprises*. Some interesting observations related to accelerators reported by the authors in [15] were that, after graduating from accelerator programs, startups were able to find investment and create profit; furthermore, more than 70% were still running after participating in the program.

> Incubators focus on incubating general business ideas, whereas accelerators focus on accelerating business ideas that have the potential to scale fast. Both organizations can work together, whereby scalable business ideas identified through incubators can be accelerated through accelerators.

2.3 Co-working Spaces

Incubators and accelerators need to provide startup companies with infrastructures (i.e., co-working spaces), which can include rental offices (including tables and chairs), organizational support (such as WLAN connections, printing), and conference meeting support [4, 14]. The co-working space needs to fulfill the following criteria: *access to information, access to knowledge, access to symbolic [i.e., important] resources, access to social capital, and opportunities for serendipity* [12]; assist with providing a place to perform events and programs; and provision of a work location for startups to develop the business idea into a prototype and full-fledged product [8]. Furthermore, six types of co-working spaces are *public offices, third places, collaboration hubs, co-working hotels, incubators, and shared studios*, as discussed in [10], and which were characterized based on *business model (for profit and nonprofit)* and *user involvement (public, semiprivate, and private).*

In the Oulu startup ecosystem, some organizations, such as Business Kitchen, Kielo, and Njetwork Inn,[12] were providing co-working spaces. Njetwork Inn's objectives were to provide infrastructure and mentorship to their tenants, build cooperation among startups (tenants), and form relationships so that some startups could offer products/services or work together. As interviewee C14 described about co-working space:

I think the atmosphere here is relaxed and all the companies, they help each other. For example we have had customers from Kielo, from other companies, paying customers and when you need help there are other companies you can ask some kind of guidance and we have been also helping and coaching other companies here.

The Business Kitchen provided a co-working space to help people work together, which could lead to the development of teams and prototypes [C18]. The co-working space offered by the Business Kitchen is given to the participants in Avanto's program [C11]. Furthermore, the incubator Kielo also offered a working space to startup companies. Some startups [C3] did not require a co-working space while other startups [C8] avoided a co-working space as they preferred confidentiality and a quiet area.

A co-working space enables the creation of a culture where startups can collaborate to become partners.

2.4 Mentors

Qualified startup coaches and mentors need to spend a significant amount of time during the early stage to provide the necessary support and expertise to

[12]Njetwork Inn: https://www.njetworking.com/en/njetwork-inn-home-company/.

inexperienced entrepreneurs [C3], which could help with and lead to the creation of an ecosystem systematically [8]. As interviewee C02 mentioned:

They are really important because they can give the knowledge and give access to specific networks which is really important in the early stage of a startup so that you get connected to the right people and get the right knowledge so that you can build the best service or product or software, whatever your startup is about.

Mentors need to be familiar with the startup and their environment, open to new ideas, and able to create an entrepreneurial mindset among the startup founding members [C02]. Furthermore, as discussed in the previous sections, for both incubators and accelerators, the first service is coaching; hence, mentors are an essential part of their programs, who share their knowledge and expertise with program participants during mentor meetings and seminars [4]. For example, incubators such as Business Kitchen and Kielo offer mentorship and coaching to early-stage startups to help with the development of a business model, a development strategy, and making connections [C14]. A co-working space also provides mentorship to their tenants, and managers also sometimes mentor for other accelerator programs [C04]. Also, accelerators and incubators have created strong connections with local startups and entrepreneurs, which can act as mentors during their programs [C11].

According to the interviewees [C10,C14,C16], mentors can assist in the following areas:

– Marketing strategy
– Pinpointing areas where a startup needs further improvement
– Providing support in proper decision-making since founders cannot make all decisions on their own
– Sharing knowledge on how to acquire funding

Moreover, experienced founding team members did not need mentorship, especially if they had previous startup experience. Also, for some startups [C07,C09], markets and customers were directing their decisions. Furthermore, in one startup [C10], the founder had more expertise in product and business development; thus, he acted as a mentor to his team members. Some startups [C14,C15,C16] only used the infrastructure as they had experience and expertise; therefore, they did not require mentorship. Thus, it appears that mentorship is more useful to people who do not have startup experience.

Mentors can be successful entrepreneurs, and startup companies in the region can act as coaches during incubator/accelerator programs.

2.5 Events

Events provide an opportunity for entrepreneurs to network with founders of successful startups, growth-stage startups, investors, and large companies [8]. There are plenty of events in the Oulu startup ecosystem, and the most famous one is Polar Bear Pitching,[13] which provides a good channel for startups to connect with investors, media, and new experts [C02]. During the event, founding team members can ask for advice on their problems, brand their product, build trust with potential customers, and pitch their idea to attract new investors. As interviewee C04 described:

Events can help as well because the more you meet particular people on different kind of events again and again will build you the trust. Events are there to create the initial contact, but the trust is what you need to build by consistently, going to these events, consistently being in touch with your potential client and build that relationship further, so the events are important

A pitching event also informs participants about the latest information on startups, and provides startups with global feedback and visibility [C03,C04,C11].

Avanto's accelerator program ends with a significant public event called Demo Day,[14] in which participants are asked to present and pitch their ideas to investors and startup experts to get their feedback. Similarly, Njetworking also conducts an event called Socializing Friday,[15] which usually occurs six times a year to enable networking among the people. Another example of this event in the Oulu startup ecosystem is Startup Weekend Oulu.[16]

Events can enable startups to brand themselves and to get feedback from experts and future investors.

3 Related Work and the Research Method Used in the Study

3.1 Related Work

Regarding the startup ecosystem, earlier studies such as [5, 8, 11, 17] discussed these aspects. For example, [8] talks about a startup and innovation ecosystem in one Australian city, which provides a model on how to create a startup and innovation ecosystem environment. Similarly, [5] discussed New York City, in which various actors were analyzed concerning the startup ecosystem. Also, in our earlier work [17], we conducted a systematic literature review on startup ecosystems. Regarding previous studies on incubators, accelerators, and co-working spaces, studies such as [3, 4, 10, 14] distinctly discussed some of these aspects.

[13] Polar bear pitching: https://polarbearpitching.com/.

[14] Demo day: https://www.businesskitchen.fi/en/events-list/2018/4/24/avanto-program-demoday.

[15] Socializing Friday: https://www.facebook.com/SocializingFriday/.

[16] Startup weekend Oulu: https://www.facebook.com/startupweekendoulu/.

3.2 Research Method

The supporting factors such as incubators, accelerators, co-working spaces, mentors, and events, which were discussed in Sect. 2, were studied in a case study of an Oulu startup ecosystem [16]. During the case study, case units such as startups, incubators, accelerators, co-working space organizations, and venture capitalist firms in the Oulu startup ecosystem were analyzed by collecting data through convenient sampling by conducting interviews with founders, mentors, and business developers working in case units. More information on case units and interviewees' details are shown in Table 2. During the interviews, an interview script was used, in which one section was set to explore the roles of the supporting factors in the startups of the Oulu startup ecosystem. A professional company later transcribed the interview recordings. Interview transcripts were analyzed using NVivo (a qualitative analyses software) with the use of deductive coding technique to extract information with respect to aspects such as incubators, accelerators, co-working spaces, mentors, and events. The results of the analyzed data are discussed in Sect. 2.

Table 2 Case units and interviewees' details

Id	Case units (C)	Size	Interviewee role
C01	Startup	1–5	Founder/SW developer
C02	Accelerator	1–10	Business developer/Mentor
C03	Startup	1–4	Founder/CEO Founder/CTO
C04	Co-working space	1–5	Event manager
C05	Venture capitalist	1–5	Investor/partner
C06	Startup	1–5	Founder/CEO
C07	Startup	1–5	Founder/CEO
C08	Startup	1–5	CEO
C09	Startup	1–5	Founder/SW developer
C10	Startup	1–20	Founder/Product owner
C11	Startup	1-5	Founder/CEO
C12	Startup	1–5	Founder/SW developer
C13	Incubator	1–5	Mentor
C14	Startup	1–5	Founder/CEO
C15	Startup	1–10	Founder/CEO
C16	Startup	1–5	Founder/CEO
C17	Startup	1–5	Founder/SW developer
C18	Accelerator	1–10	Business developer/Mentor

4 Conclusion

4.1 Discussion on the Findings

As discussed in the Introduction (Sect. 1), this chapter was designed to explore the supporting factors of incubators, accelerators, co-working spaces, mentors, and events in the startup ecosystem and their roles in the startups. The findings (see Sect. 2) from our empirical study suggest that the supporting factors play crucial roles in the early stage of the startups, where the founders are aiming to identify the problem–solution fit and to create the team members to establish the minimum viable product or prototype to address product–market fit. One finding from the data analyses is that incubators (Sect. 2.1) and accelerators (Sect. 2.2) focus on students and experienced personnel, and provide co-working spaces (Sect. 2.3) along with mentorship (Sect. 2.4) to the entrepreneurs participating in the programs. They also create events (Sect. 2.5) to attract new entrepreneurs and startup enthusiasts. The difference between incubators and accelerators was that the incubator's objective was to incubate a business idea and create a mindset for the participants to recognize whether the business idea is feasible enough to scale, while the accelerator's aim was to accelerate the process of that scalable idea through intensive mentorship and training. Figure 1 shows the interrelationships between incubators, accelerators, co-working spaces, and events in the Oulu startup ecosystem as discussed in Sect. 2.

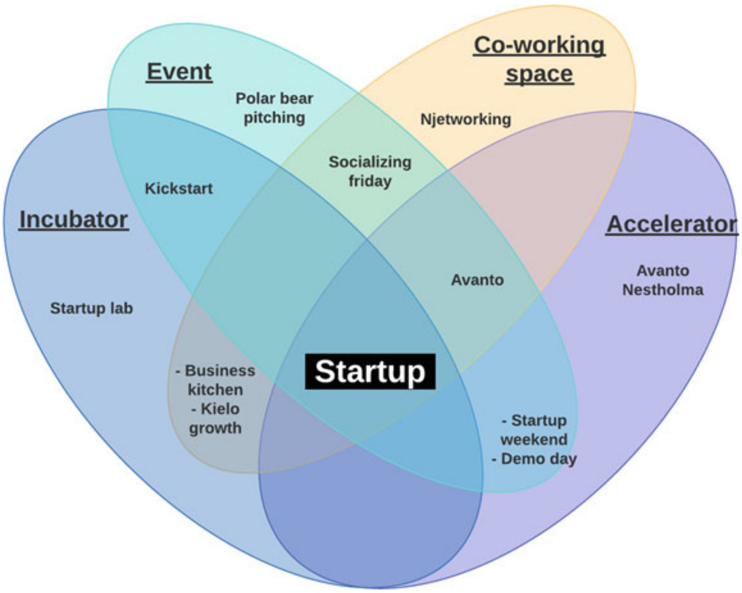

Fig. 1 Dimensions of relationships between incubators, accelerators, events, and working spaces in the Oulu startup ecosystem

For example, examples of accelerators are shown in the purple oval shape, and the events they conduct are displayed in the intersection of event and accelerator oval shapes.

4.2 Implications for Entrepreneurs and Researchers

For Entrepreneurs The points discussed in this chapter indicate that inexperienced entrepreneurs interested in starting a new venture, it can be useful for them to demonstrate their business ideas to others by participating in the incubator's program to evaluate whether the business idea is worth scaling. This process can lead to finding potential future co-founders and team members to work further on the business idea toward product and customer development. Furthermore, entrepreneurs can get intensive training from the accelerators, knowledge on different aspects during product and customer development from mentors, possible network opportunities with potential investors, and a place to work on the business idea through co-working spaces. Furthermore, in Fig. 2, we also described incubators, accelerators, co-working spaces, mentors, and events effects during the startup development stages (stages were adapted from Startup Commons[17] which is a recognized startup website).

For Researchers This chapter provides strong empirical evidence through 18 interviews concerning the supporting factors in the startup ecosystem and makes the following contributions to the literature:

- Incubators/accelerators focus on inexperienced and experienced individuals interested in entrepreneurship, developing new ventures and creating startups.
- For inexperienced individuals, university-based incubator and accelerator programs aim to create an entrepreneurial mindset among the students. In addition, students receive study credits for participating in the intensive programs.
- Mentors provide the necessary skills to develop a business idea and provide program participants with the right connections that can support their venture creation.
- For experienced individuals, profit/nonprofit-based incubators and accelerators can focus on supporting them.

[17] Startup Commons: http://www.startupcommons.org/startup-development-phases.html.

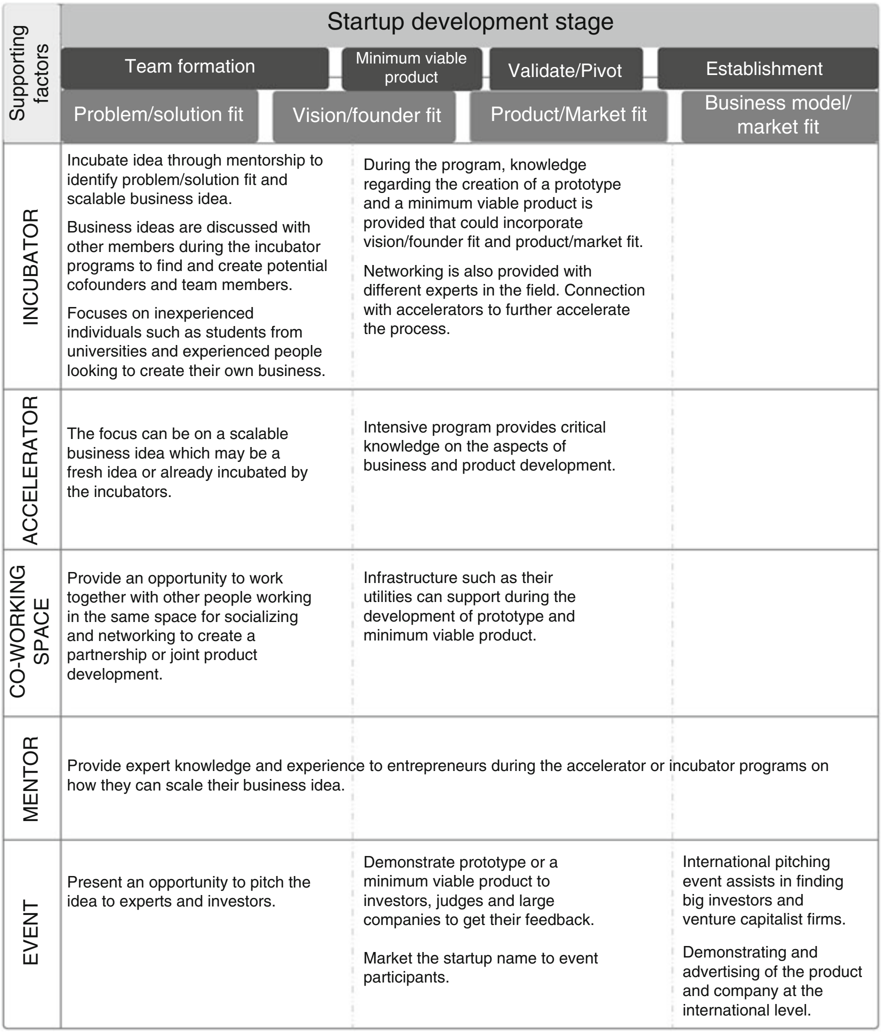

Fig. 2 Roles of incubators, accelerators, co-working spaces, mentors, and events in startup development stages

References

1. Adelino, M., Ma, S., Robinson, D.: Firm age, investment opportunities, and job creation. J. Financ. **72**(3), 999–1038 (2017)
2. Berg, V., Birkeland, J., Nguyen-Duc, A., Pappas, I.O., Jaccheri, L.: Software startup engineering: a systematic mapping study. J. Syst. Softw. **144**, 255–274 (2018). https://doi.org/10.1016/j.jss.2018.06.043. http://www.sciencedirect.com/science/article/pii/S0164121218301286
3. Bliemel, M.J., Flores, R.G., de Klerk, S., Miles, M.P., Costa, B., Monteiro, P.: The role and performance of accelerators in the Australian startup ecosystem. In: Department of Industry, Innovation & Science (Made public 25 May, 2016) (2016)

4. Cohen, S., Hochberg, Y.V.: Accelerating startups: the seed accelerator phenomenon. SSRN Electron. J. (2014). https://ssrn.com/abstract=2418000
5. Cukier, D., Kon, F., Lyons, T.S.: Software startup ecosystems evolution: the New York city case study. In: 2nd International Workshop on Software Startups (2016)
6. Davila, A., Foster, G., He, X., Shimizu, C.: The rise and fall of startups: creation and destruction of revenue and jobs by young companies. Aust. J. Manag. **40**(1), 6–35 (2015)
7. Giardino, C., Bajwa, S.S., Wang, X., Abrahamsson, P.: Key challenges in early-stage software startups. In: International Conference on Agile Software Development, pp. 52–63. Springer, Berlin (2015)
8. Haines, T.: Developing a startup and innovation ecosystem in regional Australia. Technol. Innov. Manag. Rev. **6**(6), 24–32 (2016)
9. Kane, T.J.: The importance of startups in job creation and job destruction. SSRN Electron. J. (2010). https://ssrn.com/abstract=1646934
10. Kojo, I., Nenonen, S.: Typologies for co-working spaces in Finland–what and how? Facilities **34**(5/6), 302–313 (2016)
11. Krajcik, V., Formanek, I.: Regional startup ecosystem. Eur. Bus. Manag. **1**(2), 14–18 (2015)
12. Leclercq-Vandelannoitte, A., Isaac, H.: The new office: how coworking changes the work concept. J. Bus. Strateg. **37**(6), 3–9 (2016)
13. Nguyen-Duc, A., Wang, X., Abrahamsson, P.: What influences the speed of prototyping? an empirical investigation of twenty software startups. In: Baumeister, H., Lichter, H., Riebisch, M. (eds.) Agile Processes in Software Engineering and Extreme Programming. Lecture Notes in Business Information Processing, pp. 20–36. Springer, Berlin
14. Peters, L., Rice, M., Sundararajan, M.: The role of incubators in the entrepreneurial process. J. Technol. Transf. **29**(1), 83–91 (2004)
15. Radojevich-Kelley, N., Hoffman, D.L.: Analysis of accelerator companies: an exploratory case study of their programs, processes, and early results. Small Bus. Inst. J. **8**(2), 54–70 (2012)
16. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. **14**(2), 131 (2009)
17. Tripathi, N., Seppänen, P., Boominathan, G., Oivo, M., Liukkunen, K.: Insights into startup ecosystems through exploration of multi-vocal literature. Inf. Softw. Technol. **105**, 56–77 (2019)
18. Wang, X., Edison, H., Bajwa, S.S., Giardino, C., Abrahamsson, P.: Key challenges in software startups across life cycle stages. In: International Conference on Agile Software Development, pp. 169–182. Springer, Cham (2016)

Fostering Open Innovation in Coworking Spaces: A Study of Norwegian Startups

Simone Sperindé and Anh Nguyen-Duc

Abstract Coworking spaces and open innovation are two trends that emerged in the early 2000s and have gained considerable attention. Although there exists a vast amount of research on either of these topics, the connection between them has not been much explored. The aim of this research study was to assess the state of practice of open innovation in coworking spaces and to propose a model that captures this phenomenon. Empirical data were collected by surveys and interviews with seven entrepreneurs operating Norwegian coworking spaces and two managers of coworking spaces. We found that coworking spaces express a large potential to foster open innovation among early-stage startups. Also, open innovation was found to already occur in coworking spaces: Among the four coworking space dimensions analyzed—places, spaces, events, and projects—events were regarded as the most important ones, since they act as enablers for cooperation dynamics.

Keywords Open innovation · Coworking space · Early-stage startup

1 Introduction

Coworking spaces (CWS) have become a popular phenomenon among entrepreneurs [1, 2] by creating a community, based on shared values of collaboration, openness, trust, accessibility, and sustainability [3, 4]. Since the first coworking space initiated in 2005 by Brad Neuberg, there is a rapidly increasing

S. Sperindé (✉)
University of Trento, Trento, Italy

A. Nguyen-Duc
Department of Business and IT, University of Southeast Norway, Bø i Telemark, Norway
e-mail: angu@usn.no

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_10

number of coworking spaces worldwide every year. A statistic source forecasts the number of coworking spaces in the USA to be more than 30,000 in 2020, hosting an estimated 5.1 million members [5].[1] Coworking spaces usually run for profit by entrepreneurs, and the largest companies active in the sector have become multinational enterprises. While it is true that coworking spaces provide a cheap alternative to a traditional office arrangement, people have many more reasons to choose them as their workplace. Coworking spaces connect diverse organizations and individuals, giving them the chance to collaborate, share knowledge, and develop systemic solutions to the issues they are trying to address.

The openness, collaborative culture, and the feeling of a community suggest that individuals or startups who sit in coworking spaces have a good potential to seek for external resources in a coworking space. However, the role of coworking in facilitating interorganizational collaboration is controversial [2]. In a scenario, coworkers can work on their own objectives, circulating their own networks of resources. In another one, coworkers can collaborate by providing and consuming value beyond the companies' boundaries.

As one of the known threats to early-stage startups, entrepreneurs often find a lack of connection and useful contact. Coworking spaces provide opportunities for interaction and collaboration, boosting entrepreneurial self-efficacy [1]. However, Parrino argued that mere co-location alone does not foster collaborations that lead to innovation, and a type of organizational mechanism is required for collaboration between coworking participants [6]. We are interested in understanding which setting of a coworking space can encourage the collaboration and innovation generation among entrepreneurs and startups.

Open innovation, a conceptual framework by Henry Chesbrough [7], suggests a way to look at inflows and outflows of knowledge to accelerate internal innovation and expand the markets for external use of innovation, respectively. Although open innovation is an extensive research area [8], there has been no attempt to adopt the paradigm in the context of startups in coworking spaces. From our research objective, we derive two research questions (RQs):

- RQ1: How do coworking spaces foster open innovation practices?
- RQ2: How should coworking spaces be organized to increase cooperation among members?

This work aims to contribute to both the literature about coworking spaces and open innovation, by proposing a novel point of view of both phenomena, which was not considered before.

[1] https://gcuc.co/

2 Coworking Spaces, Open Innovation, and Possible Grounds for Interconnection

The adoption of open innovation in the context of coworking spaces remains unexplored in innovation management research. For open innovation to be in place, a network is of crucial relevance [9]. Formal cooperation is often the result of informal talks and meetings, and knowledge gets shared through a set of actors that are somehow interconnected, namely a network. In this instance, coworking spaces come into play. Coworking spaces are open, both literally and figuratively: there are no walls and the desks are often shared by coworkers, and the information flow is facilitated by the organization of events and by common spaces. Capdevila et al. [3] define a coworking space as a localized space where independent professionals work sharing resources and are open to share their knowledge with the rest of the community. The way they might act as enablers for open innovation is easily recognized. Not only coworking spaces help lower the amount of many expenses such as office rent and electricity [10]. More importantly, the cost of accessing resources and competences held by others is much lower than it would otherwise be. Some coworking spaces tend to specialize toward one sector to host a set of coworkers with similar capabilities, so it is easier for a company to find just the right guy useful for its aims. Capdevila proposed a model to capture how innovation develops in a coworking space [10]. The four mechanisms of the model are places, spaces, events, and projects. *Places* represent physical locations where people interact. They are the basis for the generation of the so-called local buzz [11] and the starting point for the dynamics of innovation at a local level [12]. While people in places are characterized by geographical proximity, cognitive proximity is the distinctive element of spaces. *Spaces* are figurative locations [10] where people interact and share knowledge. The combination of places and spaces is an ideal permanent platform that makes people gather together and serves as the basis for the origin of innovation processes. *Projects* and *events* represent temporary platforms with the same purpose. Events contribute to the circulation of tacit knowledge by allowing the participation of actors that would normally be distant from each other. Thus, a larger community gains access and brings inputs to locally generated knowledge [10]. Projects help coordinate and integrate different knowledge backgrounds, by involving people that usually do not work together [10].

To analyze open innovation in any context, an evaluation framework is needed. Jones-Evans et al. [13] proposed key pillars of open innovation that can be used by researchers for their own surveys to assess open innovation practices [13]. The authors propose six pillars as indicators of the level of open innovation: (a) knowledge and technology sourcing activities, (b) innovation expenditure, (c) sources of knowledge, (d) human capital, (e) innovation networks, and (f) IP protection. For the purpose of this research, not all the pillars were considered. Pillar (b) was neglected, as the scope is not to evaluate companies' innovation level from the point of view of financial expenditure, so it is more meaningful to assess the innovation performance of a company by a pillar (a) only. Moreover, for the sake of simplicity for the survey respondents, pillar (e)—innovation network—was not considered: it is very similar to pillar (c)—sources of innovation—and the related survey questions were written so as to get information about both the aspects.

Pillar 1 Knowledge and technology sourcing activities: Aside from internal R&D activities, companies have the possibility to buy new technologies from external sources. The outsourcing of machinery, software, hardware, or any other form of technology is regarded as having an influence on the absorptive capacity of a company [14]. The analysis of the sources of knowledge and technology is therefore essential to evaluate a company's open innovation approach.

Pillar 2 Sources of knowledge: The origin of a company's knowledge is analyzed to understand what the company does to enrich its intellectual capital. Along with several different actors, external sources include all of the company's stakeholders.

Pillar 3 Human capital: The fourth pillar is about what drives the firm's internal innovation activity [15]. Interestingly, internal skills and knowledge, which are the metrics of human capital, are thought to be fundamental in the process of assimilating and applying externally acquired knowledge, i.e., the absorptive capacity [16]. For this reason, the analysis of human capital is relevant in the assessment of an open model of innovation.

Pillar 4 Intellectual property protection: Finally, open innovation calls for cooperation and sharing of jointly generated knowledge. However, companies should be able to protect their innovative efforts in order to extract value from them and to participate in the open innovation process. This ability is measured by looking at the number of IP protection mechanisms a company resorts to.

The combination of these two frameworks gives us a theoretical foundation, as shown in Fig. 1. Our RQs would explore the connection between places, spaces, projects, events, and each of the pillars in the open innovation framework [13].

Coworking spaces dimensions:
- ✓ Spaces
- ✓ Places
- ✓ Events
- ✓ Projects

Open innovation evaluation framework's pillars:
- ✓ Knowledge and technology sourcing activities
- ✓ Sources of knowledge
- ✓ Human capital
- ✓ Intellectual property protection
- ✓ Innovation expenditure

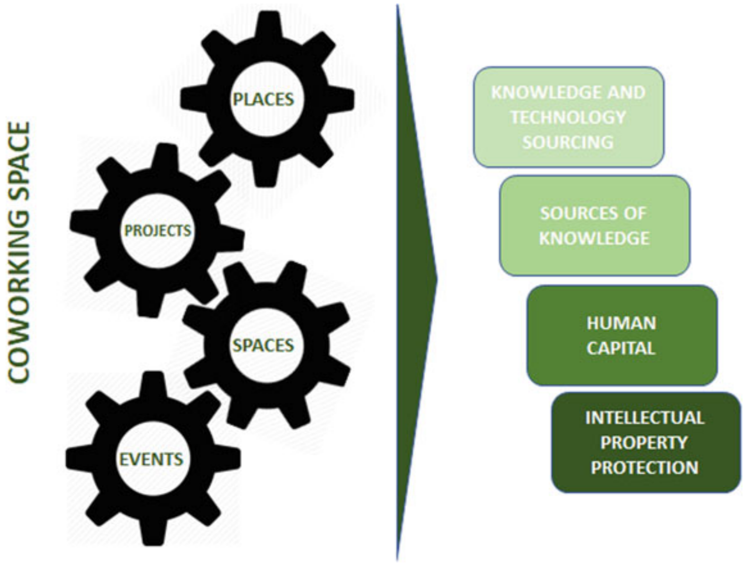

Fig. 1 A theoretical framework

3 Methods and Tools to Address the Research

This chapter adopted a survey plus interviews approach, and it targeted exclusively Norwegian companies operating at coworking spaces, without considering firm-specific variables such as size, age, or revenue. We use this approach to discover whether open innovation occurs and in which way it happens in the context of coworking. Eligible companies were identified through coworking spaces' websites and then contacted to request their participation in the study.

Although they were used to some extent to conclude, the results of the survey were not reported within this chapter, as they are out of scope. The interview results are reported in Sect. 5, through extensively explanatory excerpts.

3.1 Data Collection

The data collection was conducted in two phases, quantitatively and qualitatively, between May 2018 and August 2018. Firstly, a survey aimed at measuring the extent of open innovation happening in coworking spaces was sent to several Norwegian startups operating at coworking spaces throughout the whole country. The survey was customized from an existing one, namely the Community Innovation Survey

Table 1 Investigated coworking spaces

Coworking space	Location
Colab	Larvik
CoWorx	Kristiansand
Gowork	Asker/Drammen
Gründerhuset Hi5	Tønsberg
Gründeriet	Sandefjord
Innovation Dock	Stavanger
Ipark	Stavanger
Mediekuben	Bergen
Nordic Impact	Oslo
Oslo House of Innovation	Oslo
Startup Lab	Oslo
Validé	Stavanger
WorkWork	Trondheim

(CIS)[2] Innovasjon Norge,[3] the Norwegian Government's agency for the development of Norwegian enterprises and industry, was contacted for help in reaching out to companies. The agency kindly posted the survey on their private Facebook page, allowing us to collect more responses. The survey was sent to 230 companies based at 16 different coworking spaces in 10 cities in Norway (as seen in Table 1). The total of 37 answers was collected bringing to a response rate of 16%.

Secondly, follow-up interviews were conducted with some of the survey participants. We decided to conduct semi-structured interviews, which enabled the interviewees to contribute with their own opinions deviating from our designed questions. Ten interviews were done, but one was discarded, as it was not believed to be significant for the research. Quotes were extracted from the interviews' texts and the most significant ones are illustrated. The process is illustrated in Fig. 2, while the coworking spaces analyzed are listed in Table 2.

3.2 Data Analysis

The information obtained from the interview was analyzed qualitatively. A thematic analysis approach is chosen, following the guidelines suggested by Braun and Clarke [17]. The texts of the interviews were coded so as to spot patterns and allow an easier reporting of the data. The difference with a pure thematic analysis is that here the questions targeted some previously specified themes (namely, the model dimensions and the link between open innovation and coworking spaces)

[2] European community innovation survey: https://ec.europa.eu/eurostat/web/microdata/community-innovation-survey

[3] https://www.innovasjonnorge.no/

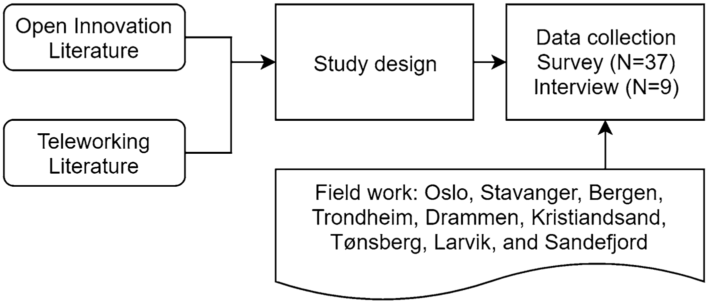

Fig. 2 Data collection

Table 2 Summary of coworkers' insights for RQ1

	Influence of coworking spaces on			
Interviewee	Knowledge and technology sourcing	Sources of innovation	Human capital	Intellectual property
I1	No	Yes	Yes	Yes
I2	No	Yes	No	No
I3	Yes	Yes	No	No
I4	NA	NA	NA	NA
I5	No	Yes	No	No
I6	NA	NA	NA	NA
I7	Yes	Yes	Yes	No
I8	Yes	Yes	No	Yes
I9	No	Yes	No	No

and no actual themes development was necessary. Rather, as the interviews were semi-structured, subthemes and quotes related to the predefined themes were sought in each of one interviewee's answers: each question was asked for examining mainly one dimension, but as the interviews were often run as conversations, quotes related to other aspects were found in several parts of the interviews. The interview transcripts were analyzed with the aid of software for text coding (Dedoose). Similar quotes were highlighted and grouped under one subtheme. Subthemes were than clustered under one of the previously established main themes. As an example, "easiness of networking" was used as a subtheme, and the main theme in this case was the value of the place dimension in the eyes of coworkers. Accordingly, the quotes under the subtheme "easiness of networking" were used for reporting within RQ1 when it came to describing coworkers' opinions of places.

3.3 Threats to Validity

While designing and conducting this study, various conscious decisions were taken to strengthen the validity of results. West et al. propose different tactics for research to increase the validity of empirical research [18].

Construct validity ensures that the interpretation of the findings is based on objective and logical criteria and that the studied dimensions are made clear and understandable to the readers. To ensure construct validity, multiple sources of evidence should be used, and a chain of evidence should be provided to the reader. Moreover, the contexts of study are various, as several coworking spaces, each with different characteristics, were investigated.

External validity refers to the extent to which the findings are generalizable beyond the context studied. To ensure external validity, the results are reported according to an existing theoretical framework [10, 13], and the initial model is refined accordingly to propose a possibility for generalization. Case study research is often criticized for external validities as it relates to the generalization of study findings. Our cases were conducted in various locations in Norway. The result of the findings can be generalized to startup ecosystems with similar context, i.e., in Scandinavian countries.

Reliability is the level to which the operational aspects of the study, such as data collection and analysis procedures, are repeatable with the same results. The present methodology section displays the reliability of the study by providing the reader with the instructions to replicate the study and thorough information about the contexts where the study was conducted.

4 Results

4.1 RQ1: How Do Coworking Spaces Foster Open Innovation Practices?

The model assumes that the four dimensions of a coworking space have a comparable influence on the development of the four pillars of open innovation. Analyzing the role coworking spaces play in promoting open innovation translates into a refined and more accurate model. Table 3 briefly summarizes the findings of the perceived influence of the coworking space on each of the specified dimensions.

Pillar 1: Knowledge and Technology Sourcing From the survey results, it was observed that internal R&D is the most practiced innovation activity. I3, I8, and I7 reported that the coworking space has an impact on their firms' internal innovation: "*[the coworking space] gives us the opportunity to test our products at events organized on purpose. "Thanks to comments people made to us, we got inspiration for trying new business models*." In these cases, the place dimension sparks

Table 3 The summary of coworkers' insights for RQ2

Suggestions for improving cooperation	✓ No suit-and-tie policy ✓ No silence policy (no forcing the coworkers to stay silent, or collaboration might not arise) ✓ Common areas separated from work desks area ✓ Noise canceling rooms ✓ A good and vibrant look ✓ Activity-based coworking
Suggestions for events settings	✓ Diversity ✓ Arrow-shaped desks ✓ Fun factors ✓ Relevant topics for the target audience ✓ Noise canceling rooms

communication. Some interviewees spoke about periodical community meetings, where members can ask for suggestions about their project. These events helped them gain useful insights.

I4, a coworking space manager, reported about some members working on a podcast, using technical tools they had never used before: according to I4 they would not have innovated their way of working, had not they shared the workplace. In this example, the presence of a permanent, ideal platform consisting of the dimensions of place and space is critical in fostering the innovative behavior mentioned: the mix between a common place where people interact with each other and a figurative community where they are located in cognitive proximity (where cognitive proximity is characterized by technical competences in the broadcasting sector that these people have in common) gives rise to a project dimension, which displays a certain degree of innovativeness.

I3 told about the way he got the opportunity to develop a digital version of his product, thanks to eight students working at the coworking space. He also said that he could gather nine people for a brainstorming session for another step in the product development, after meeting one of them at a weekly event. Such innovations clearly originated out of his company's boundaries, representing examples of open innovation. The dimensions of events and places were fundamental here.

Although of different extents, all four dimensions of coworking spaces are involved in pushing open innovation among coworking companies, which changes the way companies carry out the knowledge and technology sourcing process. The platform consisting of places and spaces gives people the chance to relate to each other in synergy. Events seem to act as connectors and to play a relevant role in bringing people to interact. Projects are the final result of the process when open innovation can ultimately be observed.

Pillar 2: Sources of Knowledge As shown in the table, all the interviewed entrepreneurs acknowledge having relied upon coworking peers for the purpose of innovation. Interviewees were asked to point out and explain the sources of innovation used within their business.

I1, I2, I7, and I8 noted that Innovasjon Norge's representatives visit coworking spaces on a regular basis and that they are available to help people solve problems, indicating that consultants represent a frequent source of innovation for coworking startups. The place dimension is clearly the enabling mechanism here: entrepreneurs get access to free consultancy just by being there.

It was observed that some coworkers try to connect with universities. I6 stated: "*We have 3 or 4 student-startups in Grunder Hub right now. As a student, you might have a lot to talk about with a 60 years old guy. It's difficult for this kind of interaction to take place in a context different from a coworking space, a place where everybody is equal.*" I7 displayed a similar way of thinking, as he explained his ideas of organizing events to involve students in his company. Projects and events spark collaboration in these two examples.

Coworking companies often cooperate on a customer–supplier relationship. As a consequence, interviewees mentioned customers as a source of innovation for their products. Likewise, competitors were highlighted as a source of collaboration. Such actors work in cognitive proximity, making the space dimension recognizable.

Several interviewees mentioned cooperation with other companies without specifying the relationship they have with them. In many cases, diversity is seen as beneficial, as I8 puts it: "*We can ask people that have no idea about our industry and see if they have some interesting inputs. Translators, accountants or someone working in hospitality can have an interesting point of view.*" I6 affirmed: "*Both people in similar sectors and in totally different sectors are surprised from the utility they get from each other.*" I2 confirms this fact: "*Coworkers represent a source of innovation: I can offer a lot more services in my company thanks to collaborations with other graphic designers.*"

Sitting in the same place provides startups with access to many opportunities for innovation. However, events are a necessary factor, as they have the power to get people to know each other, potentially fueling later collaborations.

Pillar 3: Human Capital The table reports that only a few coworkers have their core skills improved thanks to the coworking space. The survey results showed quite a high degree of competences among coworking companies: it can be inferred that coworking startups had their skills developed before or outside the coworking environment. However, in a number of cases, coworkers admitted they had the chance to learn something new thanks to their workplace.

I1, for example, said: "*It is very good to discuss personal challenges, to have someone to share it with who understands you.*" Later on in the interview, she added: "*In my office there are a lot of people close to my business, facing the same challenges, and I can learn a lot just by talking to them.*" I7 did not have his technical capabilities developed in the coworking space, but he recognized: "*If we were to start our activity from here, we would have people help us understand what skills we need to develop.*" Other entrepreneurs mentioned the fact that they see an improvement in their soft skills thanks to the coworking space, through events and other social gathering occasions.

A space dimension is observable: I1 mentioned how she could get access to knowledge by talking to people in her industry. Also, although this is not part of the absorptive capacity building process, startup managers had their soft skills improved thanks to events.

Pillar 4: Intellectual Property Protection Probably due to the fact that the interviewed entrepreneurs run small and young businesses, the last pillar was the least observed. A few examples can be illustrated.

I8 told about his encounter with a specialized lawyer during an event, who helped him apply for a patent right after. Some other interviewees said the coworking space organizes legal consultancy free of charge on a periodical basis. Events and places are the crucial dimensions in these two examples.

The Need for a Catalyst Another fact was found, especially during the interviews with coworking space managers. From their answers, it was evident how a coworking space does not do the job by itself. I4 commented: "*Just by having a coworking space I don't think you contribute anything to innovation. You have to create meeting places, an environment of trust*." I4's statement is supported by the birth of the coworking movement itself: if the place did everything needed to stimulate synergies, then serviced offices or telecenters would be the optimal solution, and coworking spaces would not have been born. I4 added: "*In the café, on a day to day basis, you have a lot of people around, I'm not just going to sit and talk to you, I don't know who you are. But once I know you because we met at an event last week, then that barrier is overcome*."

People in a coworking space will not necessarily meet, and they have different schedules and to-dos. I4's firm belief is that there is the need for a catalyst: a person that "*listens to the vibration and the chatter on the surface and puts people together*." This was confirmed by I8: "*There's the need for someone who pushes collaboration. You should have a connector in coworking spaces, someone who knows everybody and connects people.*" Unsurprisingly, I9 expressed his positive feelings about the activity of the management of his coworking space. He mentioned a very active Facebook page with something happening every day. I4 asserted that "*after a while it will go by itself and I think that's what happened in [a coworking space] in Stockholm. They started with a lot of events, like Friday meetings with free beer and so on, but part of the events was always dedicated to some kind of learning or sharing or pitching. Now big companies there sit side by side with teenager YouTubers and they respect them, because they have some knowledge that corporates lack*." In addition, I6, who defines himself as community and events manager at his coworking space, explained: "*Innovation needs people meeting and talking and helping each other: eventually they start a collaboration. It's a sharing experience, a network. People here get to know each other because there are a lot of social activities, starting from lunch together*."

Summing up, for a coworking space to foster open innovation practices, a catalyst is thought to be of crucial importance. The catalyst uses events to build a community and to connect people. Events were observed to originate connection among people, sparking the influence of places, spaces, and projects on the four pillars of open innovation. In contrast with the model proposed previously, the coworking space dimensions have different levels of influence on the four pillars.

4.2 RQ2: How Should Coworking Spaces Be Organized to Increase Cooperation Among Members?

Answering the second research question helps to understand what tools the catalyst should use and how the space should be organized to increase the chances for open innovation practices to arise within a coworking space.

Collaboration-Enhancing Features One aspect positively underlined by many interviewees (I2, I7, and I9) was the informal atmosphere present in their workplace. Suit-and-tie policies, for example, might make people feel unrelaxed and prevent them from starting a chat by the coffee machine. Another often cited aspect is the presence of common rooms, separated from the space for offices. Common areas are usually furnished with couches and tables, and they serve the purpose of interaction among people, which makes it easier for them to interact freely, being sure not to disturb someone's work. Positive comments on common areas were reported by coworking space managers. However, coworkers would in some instances prefer to have a private room, to have some privacy in the case of a confidential phone call. I7 highlights an important feature that common areas should necessarily have: "*Noise should be canceled in open rooms, by using pads, angles and furniture. This is especially important in the case of an event. Sound is really important in improving the coworking experience*." Coworkers and managers would prefer common areas to be located at the entrance, for giving a good first impression to those who enter.

A general claim about the physical appearance of coworking spaces was made by I4: "*The cool factor is quite important: it has to look good, it must have attraction power. We're done with the days where we put three engineers in a dirty old garage. If people feel 'this is the place where I want to be,' they'll put more energy!*" More energy could translate to more willingness to collaborate and innovate, so the aesthetic factor gets relevant.

When asked about his opinion of his coworking space for collaboration, I7 mentioned activity-based coworking: "*Activity-based coworking is based on different rooms for different activities. If you were to have seminars, you would use [the common room], if you were to have phone calls, you would use small rooms on purpose, if you were to have a collaboration among two or three people, you could sit on these noise-cancelling tables similar to those in an old American diner, if you want to relax and have a coffee you could sit in an open area, where you mentally invite people to sit with you and talk to you*." Activity-based coworking is an interesting approach that might be appreciated by many coworkers, as this quote testifies. It is a setting that might result in more interaction and collaboration among members and, unsurprisingly, it is already being adopted by many coworking spaces.

Fig. 3 Clockwise: a common room; inspirational quotes to motivate coworkers; wall paintings make the space vibrant; arrow-positioned desks to allow people to interact

The Right Event Setting As explained, events are thought to be the necessary catalyst for open innovation in coworking spaces. Therefore, they should be organized carefully and in a proper setting. One important aspect is the diversity that events entail in terms of participants. The more diverse are the profiles present at the event, the more opportunities might arise for everyone. Diversity was one of the most appreciated factors by coworkers. An interesting feature was observed in the events area of a coworking space. The tables are positioned in an arrow toward the projector, as shown in Fig. 3. According to the manager, I6, "*This location is the best for a mix between presentations and social gathering. The tables are positioned in an arrow towards the screen, so that you can pay attention to the presentation, but also talk to the people in front of you and have a drink or some finger food*."

I4 explained what could boost attendance in her opinion: "*To attract people to an event, it's better if you have a bit of a name*." Clearly, it is easier for an established coworking space to reach high attendance. She also added: "*[a coworking space] had its events in the shittiest places, but they gave free beer and they pulled in hundreds of people*." This underlines the necessity of a fun part beside the core of the event. Finally, she said: "*If the event itself has the right kind of program and atmosphere, it will work: people will go anywhere to hear a good speaker*." Therefore, the organizer should make sure the focus of the event is relevant to the coworking space members and to the target audience in general. Finally, the sound aspect was brought up by another coworker, namely I5: "*There are rooms where it gets really loud when everybody speaks. The events rooms should not be those ones*." Especially in the case of an event where people are expected to interact, like a workshop, it is important that the noise is canceled, or it might be hard for people to talk to each other and eventually get uncomfortable. Table 3 sums up the suggestions.

5 Discussion: Different Influences for Different Dimensions

5.1 The Refined Model of Open Innovation in Coworking Spaces

Understanding the role of coworking spaces in supporting open innovation leads to a refinement of our initial theoretical framework. Our proposed model on Open Innovation in Coworking Spaces (OICS in short) is shown in Fig. 4. We believe that events have an influence more on the development of the other coworking space dimensions, rather than directly on the four pillars (the black arrows in the figure illustrate these relationships):

- Events give the chance to people to connect. Members get to know each other better, and exchange mutual information about each other's business. Events generate spaces by putting people in cognitive proximity.

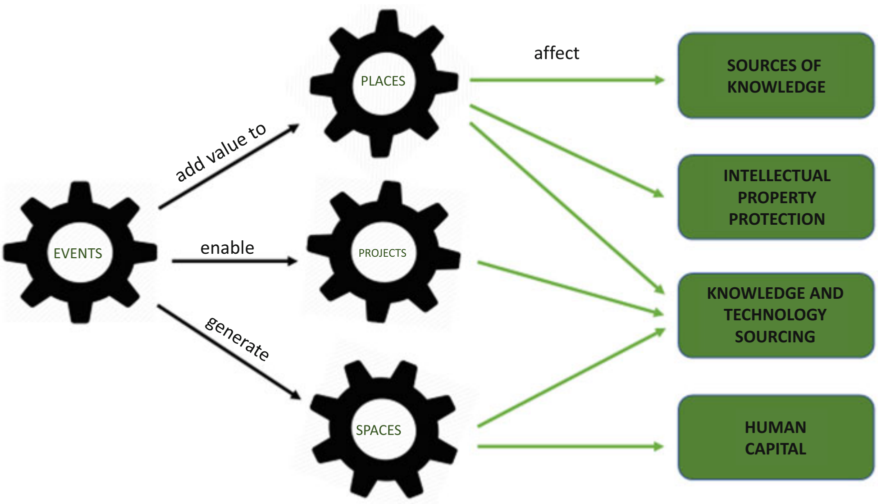

Fig. 4 The OICS model on open innovation in coworking spaces

- A common place with no mutual interaction and synergies is of low value. Events generate spaces, which then thrive within places. Events turn out to have an indirect positive influence on places, in that through the generation of spaces, they add cognitive proximity to merely geographical proximity.
- Events enable projects as they put people together.

The three dimensions shaped by the action of events (places, spaces, and projects) affect the four pillars of open innovation differently. The green arrows illustrate the relationships that we found. We explained the reasoning in the results section.

5.2 Contribution to Theory

This research's main focus was on the role of coworking spaces in fostering open innovation. While Lee et al. acknowledge the necessity of a network for open innovation practices to be in place, they do not explain the process that brings to the creation of the network [9]. This work proposes a theoretical model of Open Innovation in Coworking Spaces that explains how different elements of coworking spaces contribute and enable which level of open innovations.

This chapter fills the gap by illustrating the role of events. Capdevila compares coworking spaces to industrial clusters, by recognizing a number of common aspects [19]. However, only one out of four coworking spaces where interviews were conducted displayed a degree of specialization throughout the whole shared office and could therefore be included in Capdevila's definition. Nevertheless, specialized knowledge was observed to some extent in many coworking spaces, and members said they found it beneficial for their business. Moreover, the interviewee sitting in the specialized coworking space reported to prefer such a setting over diverse environments experienced before, since much more valuable cooperation can arise with people working in the same sector. Hence, it is believed that homogeneous shared offices can add value in terms of collaborative endeavors, but the analogy between coworking spaces and microclusters seems to be applicable to a few cases. After all, this is an obvious consequence of the fact that a large majority of coworking spaces are small and young startups themselves. Targeting only a small specific group of workers can be very risky in such a situation: for the sake of financial statements, managers are forced to accept anyone willing to join the coworking space, and specialization is a rare occurrence. In contrast, diversity was reported as positive by many other interviewees, confirming the view of Johns and Gratton [20]. Therefore, it can be inferred that, although in different ways, both diverse and homogeneous coworking spaces have the potential to foster collaboration and consequently open innovation practices.

5.3 *Opportunities for Future Research*

The model proposed in this work is based on what is observed in Norwegian coworking startups. It is not a given that the results are generalizable for each and every country. Further research should try to confirm or to validate the results by repeating the study in other geographical areas. In addition, to allow even further generalization, future research should control for different variables, which were not taken care of within this thesis: the size and age of companies, the number of members in the coworking spaces, and the background of the companies' managers, just to mention a few examples, are variables that can affect the results. Similarly, different cities might interact differently with coworking spaces. For instance, as was suggested by I4, a coworking space manager interviewed, a bigger city might fuel innovation dynamics in a coworking space, as it contributes to fill it with new people on a regular basis. In a smaller city, innovation in a coworking space is more difficult to generate, but the city benefits from it. Further research should try to find relationships between the characteristics of the city and the innovation dynamics within coworking spaces.

6 Conclusions

The hype around the coworking movement has mounted considerably throughout the last decade without signs of deceleration. Possibly, the financial crisis has played a role in stimulating the coworking phenomenon: sitting in a coworking space helps save money, and workers might have chosen this solution over other workplaces in an attempt to save on key resources during the economic downturn. Open innovation occurred recently but also became a global phenomenon across various industries. In line with the ideology of the sharing economy, companies often find it more convenient to undertake joint projects for innovation, knowing that all the actors active in the project will gain their share of value.

The academic literature had thus far neglected the link between open innovation and coworking space. This work aims at filling this gap and positions itself as one of the first research work exploring the empirical connection between open innovation and coworking spaces. The results have shown that coworking spaces have the potential to foster open innovation practices among their members. The role of events and similar initiatives is important in shaping a community. Two of the studied open innovation dimensions are majorly influenced by the role of coworking spaces, namely knowledge and technology sourcing and sources of knowledge. On the other hand, human capital and intellectual property protection are not affected as much.

Coworking space managers and companies can find much useful information in this chapter. More awareness is raised about the relationship between coworking and open innovation. This study helps companies decide on their corporate policies in terms of working arrangements and innovation practices and it advises managers on how to improve internal cooperation among coworkers. Many aspects were not considered within this study, leaving room for further development of the topic, while tracing a possible pattern for a literature flow. Future research on this theme should consider the suggestions proposed herein.

References

1. Cabral, V., Winden, W.V.: Coworking: an analysis of coworking strategies for interaction and innovation. In: Regional Studies Association Annual Conference, Graz (2016). https://doi.org/10.13140/RG.2.1.4404.5208
2. Spinuzzi, C.: Working alone together: coworking as emergent collaborative activity. J. Bus. Tech. Commun. **26**, 399–441 (2012). https://doi.org/10.1177/1050651912444070
3. Capdevila, I.: Different inter-organizational collaboration approaches in coworking spaces in Barcelona. SSRN Electron. J. 1–30 (2014). https://doi.org/10.2139/ssrn.2502816
4. Waters-Lynch, J., Potts, J., Butcher, T., Dodson, J., Hurley, J.: Coworking: A Transdisciplinary Overview (SSRN Scholarly Paper No. ID 2712217). Social Science Research Network, Rochester, NY (2016)
5. GCUC.: https://gcuc.co/2018-global-coworking-forecast-30432-spaces-5-1-million-members-2022/
6. Parrino, L.: Coworking: assessing the role of proximity in knowledge exchange. Knowl. Manage. Res. Pract. **13**, 261–271 (2015). https://doi.org/10.1057/kmrp.2013.47
7. Chesbrough, H.W.: The era of open innovation. MIT Sloan Manage. Rev. **44**, 35–41 (2003)
8. West, J., Bogers, M.: Leveraging External Sources of Innovation: A Review of Research on Open Innovation (SSRN Scholarly Paper No. ID 2195675). Social Science Research Network, Rochester, NY (2013)
9. Lee, S., Park, G., Yoon, B., Park, J.: Open innovation in SMEs—an intermediated network model. Res. Policy. **39**, 290–300 (2010)
10. Capdevila, I.: Co-working spaces and the localised dynamics of innovation in Barcelona. Int. J. Innov. Manage. **19**, 1540004 (2015). https://doi.org/10.1142/S1363919615400046
11. Bathelt, H., Malmberg, A., Maskell, P.: Clusters and knowledge: local buzz, global pipelines and the process of knowledge creation. Prog. Hum. Geogr. **28**, 31–56 (2004). https://doi.org/10.1191/0309132504ph469oa
12. Rantisi, N.M., Leslie, D.: Materiality and creative production: the case of the mile end neighborhood in Montréal. Environ. Plan. A. **42**, 2824–2841 (2010). https://doi.org/10.1068/a4310
13. Jones-Evans, D., Gkikas, A., Rhisiart, M., MacKenzie, N.G.: Measuring open innovation in SMEs. In: Vanhaverbeke, W., Frattini, F., Roijakkers, N., Usman, M. (eds.) Researching Open Innovation in SMEs. World Scientific, London (2018). https://doi.org/10.1142/9789813230972_0013
14. Doloreux, D.: Use of internal and external sources of knowledge and innovation in the Canadian wine industry. Rev. Can. Sci. Adm. [Can. J. Adm. Sci.]. **32**, 102–112 (2015). https://doi.org/10.1002/cjas.1312
15. Mariz-Pérez, R.M., Teijeiro-Alvarez, M.M., García-Alvarez, M.T.: The relevance of human capital as a driver for innovation. Cuad. Econ. 68–76 (n.d.)
16. Cohen, W.M., Levinthal, D.A.: Absorptive capacity: a new perspective on learning and innovation. Adm. Sci. Q. **35**, 128–152 (1990). https://doi.org/10.2307/2393553
17. Braun, V., Clarke, V.: Using thematic analysis in psychology. Qual. Res. Psychol. **3**, 77–101 (2006). https://doi.org/10.1191/1478088706qp063oa
18. Yin, R.K.: Case Study Research: Design and Methods. Sage, Thousand Oaks, CA (2003)

19. Capdevila, I.: Knowledge Dynamics in Localized Communities: Coworking Spaces as Micro-clusters (SSRN scholarly paper no. ID 2414121). Social Science Research Network, Rochester, NY (2013)
20. Johns, T., Gratton, L.: The third wave of virtual work [WWW Document]. Harvard Business Review. https://hbr.org/2013/01/the-third-wave-of-virtual-work (2013). Accessed 6 Nov 2018

Startup Ecosystem Maturity and Visualization: The Cases of New York, Tel Aviv, and San Paolo

Daniel Cukier, Fabio Kon, Enxhi Gjini, and Xiaofeng Wang

Abstract A healthy startup ecosystem, an environment with a well-balanced variety of agents and supporting processes, is crucial for the development of innovative startups. However, not all startup ecosystems are equally developed, and it is difficult to have all the elements of a startup ecosystem in advanced and prolific states, especially due to the fact that startup ecosystems are dynamic and evolve over time. In this chapter, we present a maturity model for startup ecosystems, which is built upon an analysis of three major startup ecosystems—Tel Aviv, São Paulo, and New York. We also present the visualization of the maturity model enabled by a web-based application that provides a user-friendly graphical representation of the maturity of a startup ecosystem. The chapter demonstrates that the maturity model, aided by the proper visualization, can serve as a basis for stakeholders in a startup ecosystem to analyze their environment, identify weak spots, and propose policies and practical actions to improve their ecosystem over time.

Keywords Startup ecosystem · Maturity model · Visualization

1 Introduction

In the era of quick technological advancements, startups, the entrepreneurial ventures, are the new companies that are changing the world, revolutionizing entire industries with innovative solutions. Boosted by the Internet, the omnipresence of mobile devices and the abundance of cloud-based services, software startups with scalable business models are an important element in the modern economy. Correspondingly, startup ecosystems have also gained increasing significance. These

D. Cukier · F. Kon
University of São Paulo, Computer Science Faculty, São Paulo, Brazil
e-mail: kon@ime.usp.br

E. Gjini · X. Wang (✉)
Free University of Bozen-Bolzano, Faculty of Computer Science, Bolzano, Italy
e-mail: xiaofeng.wang@unibz.it

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_11

ecosystems are very dynamic environments that include many startup companies in different stages of development, various types of organizations, and people, all being in continuous interaction with one another.

Startup ecosystems are not static entities. They are similar to biological ecosystems, behaving like living organisms and changing over time. Some startup ecosystems have existed for over 50 years, while others are newly born. This difference in evolution and maturity makes comparing them a challenge. Moreover, if they are to evolve toward fruitful and sustainable environments, nascent ecosystems need a clear vision of how to develop their community.

Considering the existence of hundreds of technological clusters in different countries, it is difficult to identify what is the level of development of each ecosystem. This is why having a way to measure the maturity level of an ecosystem with respect to multiple factors would be useful for comparing different realities and also proposing practical actions that can help existing ecosystems improve.

Maturity models have been used in the software industry as a tool to assess people, culture, processes, and technologies [1]. These models define a methodology to evaluate software development companies and IT management processes. They are not prescribed processes themselves, but descriptions of the characteristics of effective processes. The application of maturity models has been widened to more than 20 other domains during the last two decades. However, classifying the maturity of a startup ecosystem in a city is very different from classifying software development processes in a company. When deciding the maturity of a startup ecosystem, it is important to analyze both its static characteristics and its dynamics, to emphasize the relationships among ecosystem agents instead of only describing the elements as isolated entities, and to map the key factors of each maturity level as well as the path to the next level [2]. With such a maturity model, it is possible not only to compare different ecosystems, but also to identify gaps and propose customized practical actions that can yield meaningful improvement and lead ecosystems to the next level of development.

In this chapter, we describe a recently developed maturity model of startup ecosystems by two of the coauthors [2]. The model was developed drawing upon an extensive empirical study of three ecosystems in three different geographical regions: Tel Aviv, New York, and São Paulo. It sheds light on the characteristics and dynamics of startup ecosystems as well as their evolution path. The description is enriched by a set of visuals rendered through a web application that implemented the maturity model.

2 A Startup Ecosystem Maturity Model

The theoretical grounding of the maturity model is a conceptual framework for startup ecosystems developed after an extensive literature review on startup ecosystems and a detailed qualitative research conducted in two existing ecosystems: Tel Aviv [3] and São Paulo [4]. Two different techniques were applied during the period of the empirical study: (1) a multiple case study involving 80 semi-structured interviews conducted with key players of both ecosystems, including entrepreneurs, educators, executives investors, etc. and (2) a systematic workshop/focus group conducted in São Paulo.

Four maturity levels of a startup ecosystem:

– M1: Nascent
– M2: Evolving
– M3: Mature
– M4: Self-sustainable

The conceptual framework, illustrated in Fig. 1, clearly represents a set of important elements that play key roles in a startup ecosystem, including Startup, Entrepreneur, Market, Legal frame, etc., and a wide number of relationships among them. The relationships in the conceptual framework have two graphical representation: (1) continuous arrows, which represent the primary relationships that exist in a startup ecosystem almost all the times; (2) dotted arrows, which illustrate the relationships that are spotted only part of the times. The type of a relationship is defined by the label positioned close to an arrow.

The conceptual framework was essential in the creation of the maturity model. By an in-depth analysis of the collected empirical data, the main elements of the conceptual framework were transformed into a list of 22 factors. The factors were categorized in two groups: essential and complementary factors. The essential factors are important to take into consideration when a specific level of maturity has been reached by the ecosystem, while the complementary ones are important in advancing the ecosystem to the next maturity level. By evaluating each of these 22 factors and classifying the possible results, a scale was created, containing for each factor three possible levels of advancement: L1, L2, or L3. Table 1 shows the list of factors and the defined scales (see [2] for a detailed definition of all the factors).

Based on the percentage of the factors at each level (L1, L2, or L3), four maturity levels were defined [2]:

- Nascent (M1): In this state, the startup ecosystem is already recognized as a startup hub, with already some existing startups, a few investment deals, and government initiatives to stimulate or accelerate the ecosystem development. However, there is still no great output in terms of job generation or worldwide penetration.
- Evolving (M2): In this state, the startup ecosystem is in the evolving stage with a few successful companies which also have regional impact, job generation, and a small local economic impact. To be at this level, the ecosystem must have all essential factors classified at least at L2, and 30% of complementary factors also at L2.

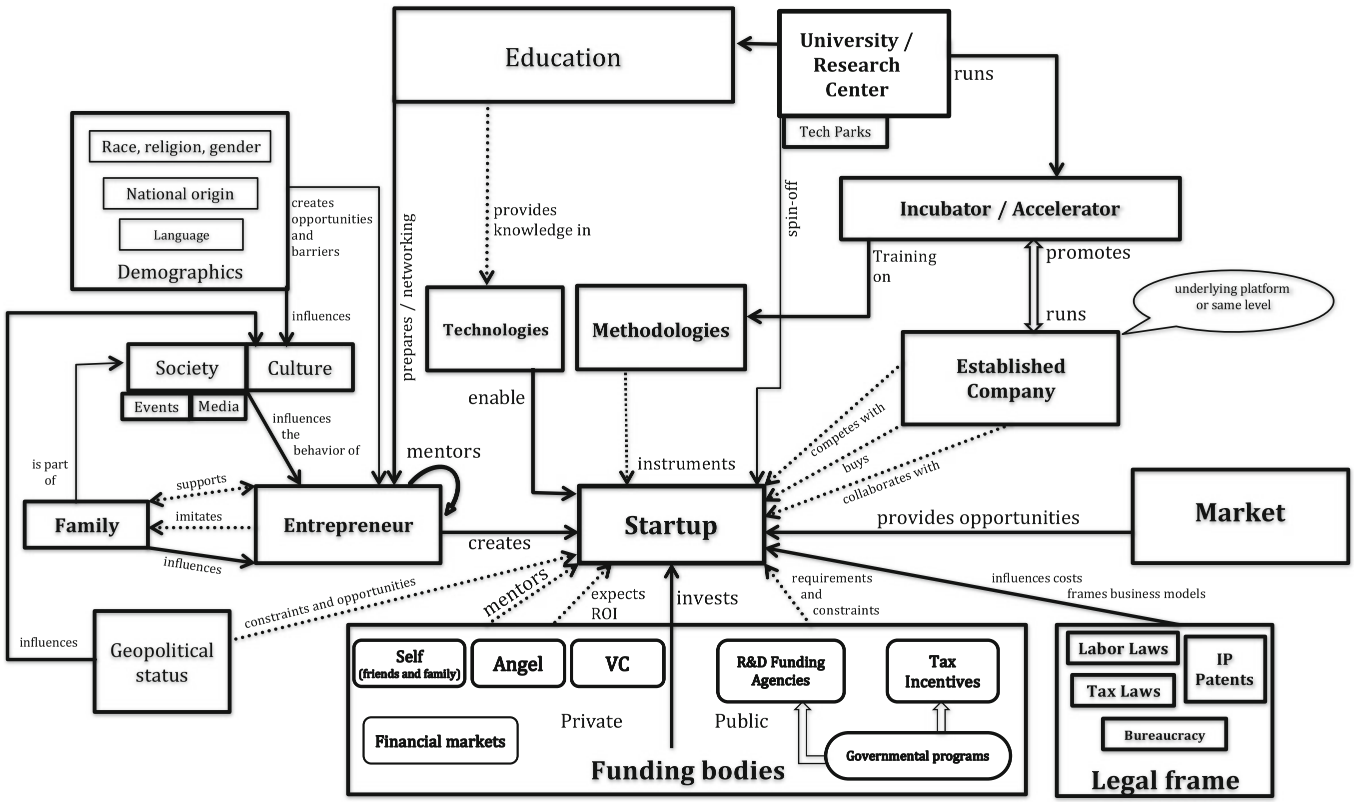

Fig. 1 The conceptual framework behind the maturity model, excerpted from [2]

Table 1 The startup ecosystem maturity model, adapted from [2]

	Level of advancement		
Factors	L1	L2	L3
Exit strategies[a]	0	1	≥2
Global market[a]	<10%	10–40%	>40%
Entrepreneurship in universities[a]	<2%	2–10%	>10%
Culture values for entrepreneurship[a]	<0.5	0.5–0.75	>0.75
Startup events[a]	Monthly	Weekly	Daily
Ecosystem data and research[a]	N/A	Partial	Full
Ecosystem generations[a]	0	1	≥2
Mentoring quality	<10%	10–50%	>50%
Bureaucracy	>40%	10–40%	<10%
Tax burden	>50%	30–50%	<30%
Accelerators quality (% success)	<10%	10–50%	>50%
Access to funding in USD/year	<200M	200M–1B	>1B
Human capital quality	>20th	15–20th	<15th
Technology transfer processes	<4.0	4.0–5.0	>5.0
Methodologies knowledge	<20%	20–60%	>60%
Specialized media players	<3	3–5	>5
Relative measured factors (per 1 million inhabitants)			
Number of startups[a]	<200	200–1k	>1k
Angel funding in number of deals/year[a]	<5	5–50	>50
High-tech companies presence[a]	<2	2–10	>10
Access to funding in number of deals/year	<50	50–300	>300
Incubators/tech parks	1	2–5	>5
Established companies influence	<2	2–10	>10

[a] Essential factors

– Mature (M3): In this state, the startup ecosystem includes hundreds of startups, where there is a considerable amount of investing deals, existing successful startups with worldwide impact, and a first generation of successful entrepreneurs who help the ecosystem to grow and be self-sustainable. To be at this level, the ecosystem must have all essential factors classified at least at L2, 50% of complementary factors also at L2, and at least 30% of all factors at L3.
– Self-sustainable (M4): In this state, the startup ecosystem includes thousands of startups and financing deals, at least a second generation of entrepreneur mentors, especially angel investors, a strong network of successful entrepreneurs engaged with the long-term maintenance of the ecosystem, and an inclusive environment with many startup events and high-quality technical talent.

Table 2 The metrics importance for each maturity level, adapted from [2]

Maturity metric	Maturity level			
	M1	M2	M3	M4
Exit strategies	a	a	c	c
Entrepreneurship in universities	c	c	b	a
Angel funding	a	a	b	c
Culture values for entrepreneurship	c	c	c	b
Specialized media	a	b	c	c
Ecosystem data and research	a	a	b	c
Ecosystem generations	a	a	b	c
Events	c	c	b	a

[a] Not so important
[b] Important
[c] Very important

The maturity model also includes a simplified metrics importance table (as shown in Table 2) which indicates how significant a factor is in each one of the maturity states. This can be helpful for the main players in a startup ecosystem by showing them where they should focus their efforts for the ecosystem's development.

The presented maturity model has been validated and refined through an extensive case study of the New York startup ecosystem.

After the maturity model validation, we developed a web-based application to provide a graphical and interactive interface to the maturity model. The objective is to represent the model in an easier to understand and usable manner, and to enable different stakeholders to explore the characteristics and status of a startup ecosystem, without the preknowledge of the maturity model underneath.

3 The Maturity of Three Startup Ecosystems

Tel Aviv, São Paulo, and New York startup ecosystems were investigated in the process of building the maturity model, and their maturity evaluations are presented in this section.

3.1 An Overview of the Three Startup Ecosystems

Tel Aviv Startup Ecosystem Israel is well known as a startup nation [5], and is reputed to have the most startups per capita. Most startups in Israel are located in

Tel Aviv, which is a highly ranked startup ecosystem with global fame. Despite the history of conflicts and tensions, Tel Aviv has become a leader in launching high-tech businesses. Being a country with a small population and a paucity of natural resources, Tel Aviv and Israel in general have to rely on alternative resources, especially people. Israel's mandatory military service is a contributing factor in this aspect, as it exposes young people to accountability and responsibility.

After certain economic difficulties in the 1980s and a change in the government toward a more liberal view of the economy, public policies changed toward strengthening the private sector. Although still with a strong well-fare state and government-supported funding for health, education, and research, nowadays, the private sector is the one that drives most of the Israeli innovation and economy. Beginning in the 1970s, large high-tech multinational companies started to establish research and development (R&D) centers in Israel attracted by the comparatively cheap, high-quality scientific and engineering labor. Nowadays, Israel hosts R&D centers from most major IT companies in the world, including Intel, IBM, Microsoft, Google, HP, Yahoo!, Facebook, Oracle, SAP, Cisco, Siemens, and Motorola [3].

São Paulo Startup Ecosystem São Paulo is the largest Brazilian city, the 12th largest city in the world. It is the financial center of Brazil and hosts the headquarters of many major companies and banks, including many foreign companies doing business in Brazil. São Paulo is home to the Bovespa, the largest stock and bond exchange in Latin America. It has several leading science and technology universities. Foremost among them is University of São Paulo (USP), founded in 1934, one of the largest universities in the world, with more than 90,000 students (of whom almost one-third are masters and doctoral students); 4 of its 11 campuses are located in the São Paulo metropolis [4]. The city concentrates over 60% of startup investments in Brazil and well over 2000 ventures working on tech-based products and services.

São Paulo is the Latam base of many Fortune 500 companies and tech giants like Spotify, Airbnb, Google, Netflix, and Amazon. It is also home to many local unicorns, including 99 (sold to Didi), PagSeguro (IPO at NYSE), Nubank (now a decacorn valued at US$10B), Stone (IPO at NASDAQ), and iFood.

New York Startup Ecosystem New York City is the business capital of the world, as well as the center of advertising and the financial, food, and fashion industry. It is supported by a robust high-tech entrepreneurial policies system and a strong pool of human capital, blossomed into FinTech, FashionTech, FoodTech, AdTech, Marketing Tech, Real Estate Tech, and so on.

In the late 1990s, New York City was in its nascent phase and had already acquired much of the necessary support infrastructure to evolve quickly: the metropolitan region is home to top research universities like Cornell, Columbia, New York University, and the City University of New York, which all have special programs for entrepreneurs; many (sometimes free) co-working spaces like General Assembly and WeWork (which was valued $17 billion in 2016) started to emerge; the public transportation system is efficient; and big tech companies established offices in the city (for instance, Google's office in the Chelsea neighborhood).

3.2 The Maturity Levels of the Three Ecosystems

Applying the maturity model described in Table 1, the maturity levels of the three ecosystems were evaluated as shown in Table 3.

The analysis of the three startup ecosystems at two different maturity levels, and their evolution through the maturity levels, revealed many insights regarding how a startup ecosystem can evolve healthily. Among them, several key points are:

The Minimum Requirements for a Startup Ecosystem to Exist in Its Nascent Stage One of the first requirements for an ecosystem to exist is to have great entrepreneurs. It seems obvious that any startup ecosystem needs entrepreneurs, but it is not so obvious that the entrepreneurs are the seed of everything. This means that talented entrepreneurs are necessary even at the first nascent stage of an ecosystem. The existence of high-quality research universities in the region is an important attractor for these talents, especially when there are programs for tech entrepreneurship. The presence of big tech companies can also be considered a talent attractor, but not necessarily all the talents will become entrepreneurs.

By analyzing the three startup ecosystems, it is clear that all of them surpassed the nascent stage.

The Requirements for a Startup Ecosystem to Be Self-Sustainable A startup ecosystem reaches a self-sustainable level when there are at least two generations of successful entrepreneurs that start reinvesting their wealth in the ecosystem by becoming angel investors and offering their mentorship. This is only possible when there are many opportunities for merge and acquisition (M&A) as well as initial public offerings (IPOs) in the market, and, moreover, when the entrepreneurial culture is widely accepted and understood, supported by high-quality educational institutions, and startup events happen almost every day. When the ecosystem reaches the self-sustainable maturity level, the media also plays the role of maintaining the momentum and awareness of the public.

Both Tel Aviv and New York are considered to have reached the self-sustainable maturity level. On the other hand, São Paulo has not reached this stage yet, since

Table 3 The detailed information on the maturity levels of the three startup ecosystems, adapted from [2]

Factor	Startup ecosystem		
	Tel Aviv	São Paulo	New York
Exit strategies[a]	L3	L2	L3
Global market[a]	L3	L2	L3
Entrepreneurship in universities[a]	L3	L2	L3
Culture values for entrepreneurship[a]	L3	L2	L3
Startup events[a]	L3	L2	L3
Ecosystem data and research[a]	L3	L2	L3
Ecosystem generations[a]	L3	L2	L3
Mentoring quality	L3	L2	L3
Bureaucracy	L2	L1	L3
Tax burden	L2	L1	L3
Accelerators quality (% success)	L3	L1	L3
Access to funding in USD/year	L3	L2	L3
Human capital quality	L3	L2	L3
Technology transfer processes	L3	L1	L3
Methodologies knowledge	L2	L2	L2
Specialized media players	L2	L2	L3
Number of startups[a]	L3	L2	L3
Angel funding in number of deals/year[a]	L3	L2	L3
High-tech companies presence[a]	L3	L2	L3
Access to funding in number of deals/year	L3	L1	L3
Incubators/tech parks	L3	L2	L3
Established companies influence	L3	L2	L3
Essential factors	L3(10)	l2(10)	L3(10)
Complementary factors	L2(4)	L1(5)	L2(1)
	L3(8)	L2(7)	L3(11)
Maturity level	Self-sustainable (M4)	Evolving (M2)	Self-sustainable (M4)

[a] Essential factors

the required characteristics, such as an evolved IPO market or two generations of successful entrepreneurs, do not exist in the ecosystem.

The Evolution of a Startup Ecosystem It is possible for self-sustainable ecosystems to exist if the local culture values the entrepreneurial behavior. For instance, when comparing New York with Boston, the startup ecosystem in Boston (the home of MIT) did not take off as fast as that in New York, because the local culture of Boston is much more conservative, while New Yorkers are more open to risk. Entrepreneurial culture, which is something very difficult to change in a short term, plays a significant role in the evolution of a startup ecosystem.

The maturity levels of three major startup ecosystems:
– Tel Aviv: Self-sustainable (M4) – São Paulo: Evolving (M2) – New York: Self-sustainable (M4)

Nevertheless, the evolution of São Paulo startup ecosystem shows that culture can change, though it may take time. There, the first generation of tech entrepreneurs started timidly in 2000. At that time, young people were supposed to finish their university degrees and find a job. After 15 years, the scenario changed to a culture in which being an entrepreneur is a lifestyle. São Paulo is a city with many characteristics similar to New York: a large metropolis, with millions of people (mostly first-, second-, and third-generation immigrants); a financial, advertising, and business center; and a culture of hard work, where time is money. São Paulo has all the potential to evolve from the evolving (M2) level to mature (M3) or even self-sustainable, but for that to happen, it must overcome important obstacles, like developing more policies for tech-talent attraction; reducing the tax burden; improving the law framework for company creation and closing; investing in mobility infrastructure to facilitate access to high-quality universities; and advancing the investment market.

It is important to emphasize that the maturity levels of the three startup ecosystems reported in this chapter were evaluated based on the data collected in 2015 and 2016. Tel Aviv and New York were already at M4 (self-sustainable) maturity level at that moment, and so there would be no big difference compared to their maturity levels in 2019. São Paulo, on the other hand, has much evolved in the last several years. Even though it is subject to further research, it is quite probable that the maturity level of São Paulo startup ecosystem has reached M3 (mature) level in 2019.

4 Startup Ecosystem Maturity Visualization

In this section, we present a web-based application that implemented the maturity model. It provides a graphical and interactive user interface to explore the maturity evaluation of startup ecosystems.

The web application provides three levels of representation of the maturity of a startup ecosystem:

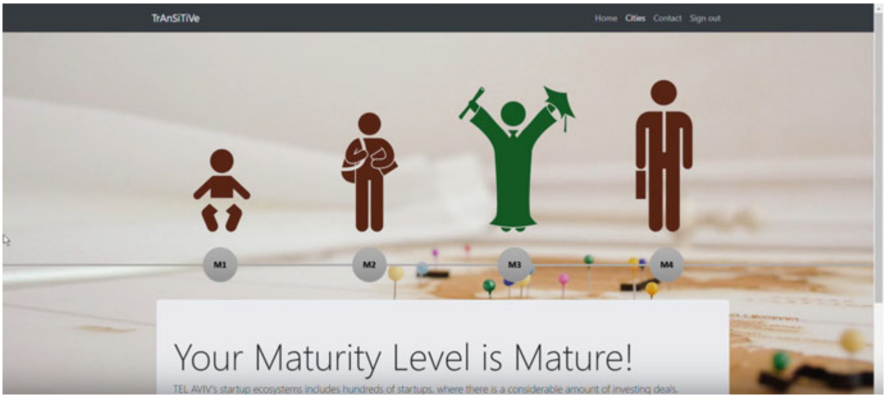

Fig. 2 The illustration of the overall maturity level of São Paulo startup ecosystem

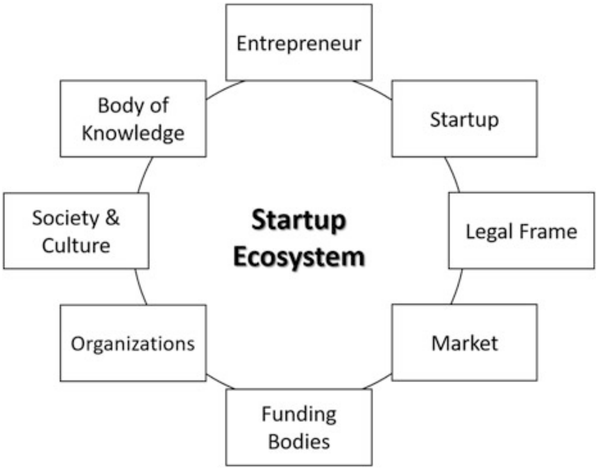

Fig. 3 The dimensions of the startup ecosystem factors

– A general view of the overall maturity level (M1, M2, M3, or M4)
– A clustered view of the maturity level along eight dimensions
– A zoomed-in view of one dimension with the clustered factors, their levels of advancement, and the explanation and advice related to that dimension

The overall maturity level of a startup ecosystem is visualized in a fairly straightforward manner, e.g., as shown by Fig. 2.

To visualize the detailed evaluation of the 22 factors and their levels of advancement (as explained in Table 3), we clustered the factors into 8 different dimensions. Figure 3 is the schema of this clustering. To visualize each dimension, we needed to decide the maturity value of each dimension, which was in turn decided by the levels of advancement of the factors mapped into this dimension. For this purpose, we used calculation tables illustrated by Table 4 for each startup ecosystem (using

Table 4 The clusters of the startup ecosystem factors into dimensions and maturity value calculation

	Dimensions							
Factors	Startup	Entrepreneur	Body of knowledge	Organizations	Society and culture	Funding bodies	Market	Legal frame
Exit strategies[a]	2			2		2		
Global market[a]							2	
Entrepreneurship in universities[a]			2	2				
Culture values for entrepreneurship[a]					2			
Startup events[a]					2			
Ecosystem data and research[a]				2		2		2
Ecosystem generations[a]		2			2			
Mentoring quality		2						
Bureaucracy							1	1
Tax burden				1				1
Accelerators quality (% success)				1				
Access to funding in USD/year						2		
Human capital quality		2	2					
Technology transfer processes			1					1
Methodologies knowledge			2					
Specialized media players					2			

Number of startups[a]	2	2					2	
Angel funding in number of deals/year[a]						2		
High-tech companies presence[a]				2				
Access to funding in number of deals/year						1		
Incubators/tech parks				2				
Established companies influence	2	2		2	2			
Evaluated maturity value of dimension	6	10	7	14	10	9	5	5
Maximum maturity value of dimension	9	15	12	24	15	15	9	12
% (Evaluated value/Maximum value)	0.67	0.67	0.58	0.58	0.67	0.6	0.56	0.42

[a] Essential factors

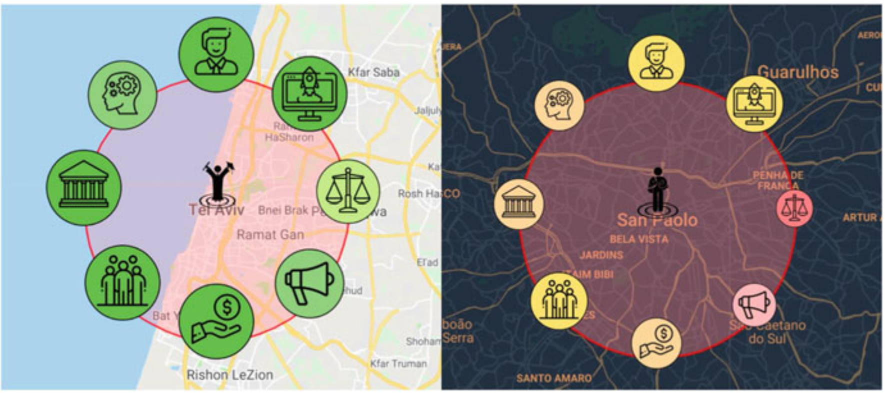

Fig. 4 The clustered view of the maturity level of a startup ecosystem: the examples of Tel Aviv and São Paulo

São Paulo as an example. Note that one factor can be mapped to more than one dimension).

The clustering dimensions and the calculation tables provide the basis to create the clustered view and the zoomed-in view of the maturity of a startup ecosystem, as shown in Figs. 4 and 5. These two views use Google Map to visualize the geographical location of an ecosystem, and also apply the day/night map style, as shown in the two figures.

We have evaluated the web application with several local startup ecosystem stakeholders in Bolzano, Italy. The results were positive from both usefulness and usability perspectives. They pointed to various ways to improve and extend the features of the application, in order to better serve the potential users, who are various startup ecosystem players, and above all, local policy makers.

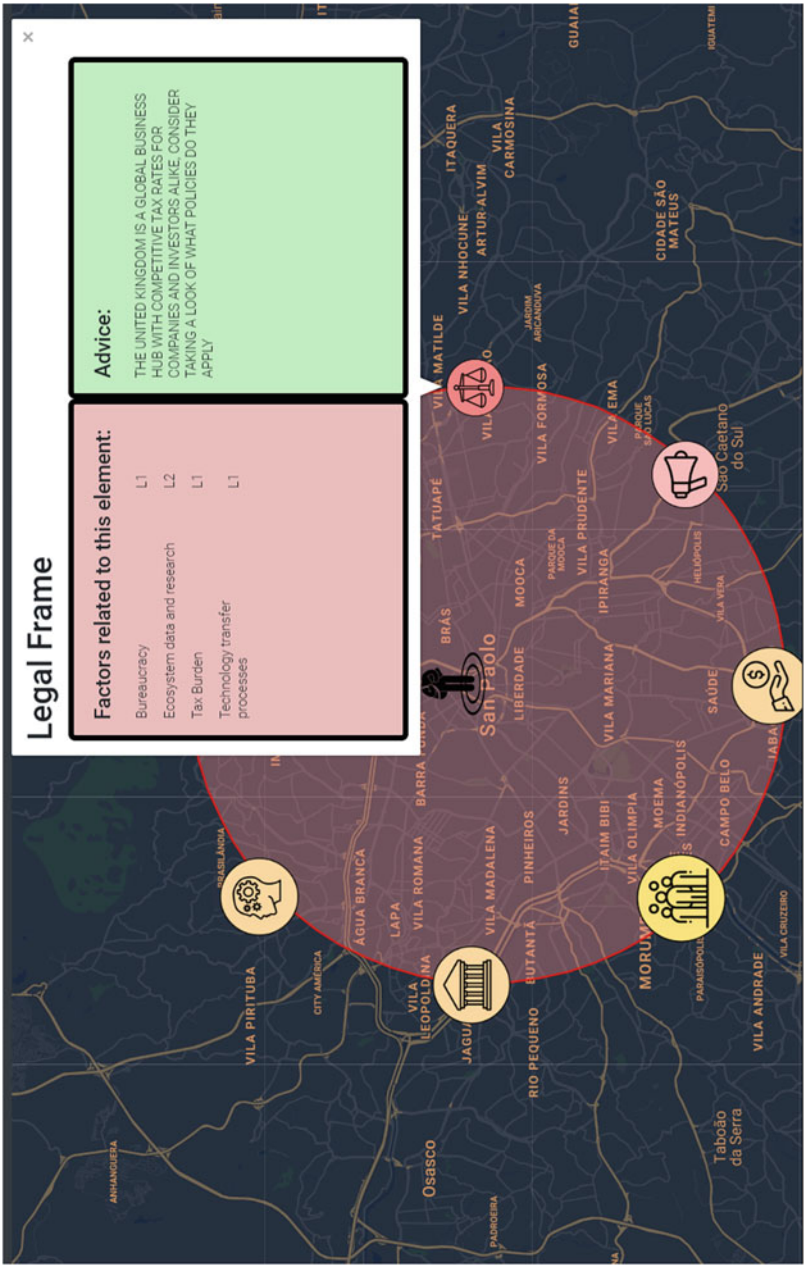

Fig. 5 The zoomed-in view of one dimension of factors from the clustered view: the example of São Paulo

5 Conclusion

A healthy startup ecosystem, an environment with a well-balanced variety of agents and supporting processes, is crucial for the development of innovative startups. However, not all startup ecosystems are equally developed, and it is difficult to have all the elements of a startup ecosystem in advanced and prolific states, especially due to the fact that startup ecosystems are dynamic and evolve over time. In this chapter, we presented the maturity model developed based on the empirical study of three startup ecosystems: Tel Aviv, São Paulo, and New York, and the visualization of the maturity model enabled by a web-based application that provides a user-friendly graphical representation of the maturity of different startup ecosystems.

Our work demonstrates that the startup ecosystem maturity model, aided by proper visualization, can serve as a basis for stakeholders in a startup ecosystem to analyze their environment, identify weak spots, and propose policies and practical actions for improving their ecosystems over time.

Future work regarding the maturity model includes collaborating with other researchers in using the maturity model to analyze new regions and derive concrete actions that should be taken to improve those ecosystems. This research could also be extended to other regions outside from big urban centers. It is a challenge to develop fruitful startup ecosystems in smaller cities. In the long term, small and medium cities tend to loose talent and resources to big centers. We consider that there is a vast field of research to be explored on startup ecosystems in small and medium cities. Regarding the implementation of the maturity model, the next meaningful step would be automating the collection of maturity-related data from various ecosystem stakeholders, which is crucial but effort-consuming, and could be streamlined using automated web-based solutions.

References

1. Mettler, T.: Maturity assessment models: a design science research approach. Int. J. Soc. Syst. Sci. **3**(1/2), 81–98 (2011)
2. Cukier, D., Kon, F.: A maturity model for software startup ecosystems. J. Innov. Entrep. **7**(1), 14 (2018)
3. Kon, F., Cukier, D., Melo, C., Hazzan, O., Yuklea, H.: A panorama of the Israeli software startup ecosystem (March 1, 2014). http://dx.doi.org/10.2139/ssrn.2441157
4. Cukier, D., Kon, F., Maital, S., Frenkel, A.: Innovation and entrepreneurship in the São Paulo Metropolis: the role of its major university (September 4, 2016). http://dx.doi.org/10.2139/ssrn.2834602
5. Senor, D., Singer, S.: Start-Up Nation: The Story of Israel's Economic Miracle. Random House Digital, Inc., New York (2011)

Thailand's Software Startup Ecosystem

Aziz Nanthaamornphong and Rattana Wetprasit

Abstract Software startups are currently very popular in Thailand, and existing information reveals an increase in the number of participants and investors in software startup businesses. Moreover, widespread events have been held to showcase the products and services these businesses have contributed. Software startups primarily develop innovations in the form of software produced from limited resources within a limited time. This software must be able to contribute to a sustainable business, and must be adjustable to each business size. Previous research indicates that both attention and emphasis must be placed on the importance of studying software startups in the form of empirical research. This will assist decision-making for those who are interested in initiating software startups and those who want to support them. Research has scarcely studied software startups in Thailand, and therefore, we are interested in Thai startups' current situation as well as the startup ecosystem. This study clarifies that software startups in Thailand are defined as newly emerging businesses anticipated to help businesses grow quickly. Each software startup is in search of a different business model, as current software startups in Thailand have been created to help and support particular businesses. However, software startups rarely invent their own unique, exotic business models or apply advanced technologies and research in their startups.

Keywords Software startups · Software engineering · Case study

1 Introduction

A software startup is a form of business with exponential growth and that can expand by searching for new business models; further, they apply technology in conducting business from newly emerged ideas. Coleman and O'Connor [3] defined software

---

A. Nanthaamornphong (✉) · R. Wetprasit
Prince of Songkla University, Songkhla, Thailand
e-mail: aziz.n@phuket.psu.ac.th; rattana.w@phuket.psu.ac.th
https://www.computing.psu.ac.th/research/aziz

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_12

startups as "unique companies that develop software through various processes and without a prescriptive methodology." Alternatively, a startup should be considered as a temporary organization, which seeks a scalable, repeatable, and profitable business model, and therefore aims to grow [2].

Currently, businesses as software startups have become immensely popular in Thailand. The public and private sectors have both financially supported this movement to open up more opportunities for those in technology startups. Moreover, several events have been held to present products from these technology startup businesses, such as the Startup Thailand Job Fest 2017 [8] and the government's Thailand 4.0 policy, the latter of which seeks to enhance Thailand's economy through innovation. Thus, many government agencies have provided budgetary support to develop software startups from around 2015 to present. Several ministries and government agencies are involved in this movement, such as the Ministry of Digital Economy and Society, Ministry of Science and Technology, and Ministry of Education. Consequently, the current software startup trend has gained significant attention in terms of the desire to initiate successful software startups, as well as both Thai and foreign investors' aspirations to invest in them.

A survey conducted from 2012 to 2017 reveals the growing number of software startup companies in Thailand [17], as well as the growing number of investors who are ready to financially support these software startup businesses. Additionally, research on software startups in Thailand collected data on all software startup businesses, their growth in funding, and information about the startup business' founders. This data indicated the growing interest in software startups in Thailand [18]. Even so, no current research has been conducted regarding the significance and ecosystem of software startups in Thailand. Moreover, software startup research is crucial during the funding process to help those involved make decisions and avoid these businesses' failure [10].

One important characteristic of the software startup is its development of innovative software, produced under pressure and with scant resources. This software must be able to contribute to a sustainable business and be adjustable to each business size [1, 7].

Previous studies demonstrate that researchers are interested in software startups due to their challenges. Specifically, companies aim to innovate with limited time and resources, and they educate themselves about software startups through such networks as the Software Startups Global Research Network [13], which is a collaboration between researchers and those interested in software startups. These networks are organized to distribute research results in the software startup context and publicize this knowledge and equipment for both operators and entrepreneurs. They also aim to reveal methods to reduce errors and increase chances of success. Although other countries have conducted partial research studies, it has been found that software startup research in Thailand is still relatively limited. Therefore, a need exists to investigate software startup businesses in a Thai context [9, 11].

The primary goal of this study is to better understand the current situation and ecosystem of software startups in Thailand. We used a research method involving in-depth interviews with Thai software startup companies, as well as the public and private sectors that support them.

2 Background

Thai startups are growing rapidly, which could be due to the accessibility of Thai technology. For instance, e-commerce startups in Thailand have earned more than USD $2.9 billion. Therefore, Thailand has the potential to promote the growth of startup businesses. However, many Thai startups have been less than successful, which might be influenced by other factors as well.

In 2017, investment in startups was mostly in e-commerce businesses (21%), compared to FinTech (9%), Logistic Tech group (8%), and e-payment businesses (8%). The smallest investment was in the food sector (6%), while the remaining investment was in other startup sectors (48%).

To understand how Thai startup ecosystems differ from other Asian countries, we examined ecosystems in Singapore, Malaysia, and China.

In Singapore, the government set up various projects to support startups. For example, in 2016, "SGInnovate" was established by a public sector under the National Research Foundation (NRF) [14] to coordinate communication between specialized business consultants, investors, and researchers. This project concentrated on startup company development, especially business groups using deep technology for commercial purposes. Examples included artificial intelligence (AI), robots, digital health, smart energy, transport, blockchain, and Big Data. In addition, SGInnovate formed a financial network between public and private sector startups, to support business plan development, seek funding, and mobilize startup companies to enter into the stock exchange. Despite a lack of financial resources, the government of Singapore realized the availability of human resources, and pushed to the world market. They formulated clear criteria concerning startup company qualifications, and facilitated joint investment between public and private sectors. For instance, to receive financial support, a startup company must be a small and medium enterprise, doing business required by the government. [16]

In Malaysia, the government had a strong objective to develop the ecosystem of FinTech, which serves the needs of financial industries, and promotes innovation in the country. Therefore, the subagency of Malaysia Digital Economy Corporation (MDEC) [4] was established to monitor and develop technology and media, as well as the country's development in the digital age. Recently, the MDEC provided startups with a centralized network, entitled Malaysia Digital Hub (MDH). This network improved capacity with high-speed internet and included high-speed fiberglass connection equipment for easy communication. Also, the MDEC linked startups with public and private investors, creating new opportunities, while facilitating interaction. In addition, the MDEC gathered networks of startup initiators across 20 countries. This allowed startups to easily expand their networks. In addition, the MDEC cooperated with leading companies such as Microsoft, Next Academy, Maybank, and Y Academy to mobilize the growth of startups. Furthermore, some programs exclusively made for technological enterprises like Malaysia Tech Entrepreneur Programme (MTEP), created opportunities for startup entrepreneurs to launch or expand their businesses in Malaysia. This channel diversified startups in the country, resulting in various FinTech services.

In China, the Ministry of Finance, and the State Administration of Taxation introduced a policy on tax assistance. This policy offered benefits for venture capital companies, angel investors, individual investors running startup companies for deep tech businesses, and eight industrial groups already stipulated by the government. Income tax reduction could be applied if income was lower than 50 million Yuan per year, and the cost ratio for research and development was at least 5% of the annual income. In addition, the cost for research and development could be deducted from the tax base for more than 150% [6]. The policy of tax deduction encouraged knowledgeable and resourceful Chinese business owners to initiate startups in their homeland. Because the Chinese believe that technology adds value, they were also interested in AI, and deep tech for both big and small businesses. Moreover, highly competitive businesses in China required technology to support an array of modern innovations [19].

3 Research Methodology

This study empirically investigated Thailand's software startup ecosystem using in-depth interviews of the government agencies, private enterprises that support Thai software startups, and Thai startup companies. The generated questions aimed to address the following research topics:

1. Contextual factors of Thai startups, such as the definition of software startup businesses, environment, culture, and investment period.
2. Support aspects such as the method of supporting startups and the criteria for selecting which startups receive support.
3. Problems and future changes in Thailand affecting startups.

We selected the interviewed participants from lists from Techsauce (2017) [17] and the New Economic Warrior platform [18]. Eventually, we selected 35 participants in Thailand, divided into two groups: (1) 5 organizations that support software startups, and (2) 30 software startups. Table 1 notes the details for the first group participants. Regarding the software startup group, businesses came from several regions across Thailand. We classified the software startups into eight groups as shown in Table 2.

Each participant was interviewed for 60–90 min, during which time data was gathered through voice recording. This information would be used in a qualitative analysis incorporating the coding analysis method [15], which is used to decrypt messages through a group format, or "theme." This is one way to divide information into portions of text from sentences, which reduces the data's complexity; moreover, it enables researchers to capture the data's key points. After the messages were grouped, we could then analyze these grouped messages to reveal conclusions on our various proposed issues.

Table 1 Organizations' profile

A001	A government agency that promotes software startups, ranging from recently launched software startups to developed software startups that want to expand their businesses. This entity opens startup incubators to support new startup businesses, and supports startups in each phase through various projects, in terms of education and funding
A002	A private enterprise that uses the company's money to invest in software startups associated with the company by opening up a startup accelerator for support. Thus, software startups can participate in the project and receive funds through investments as corporate venture capital
A003	A private enterprise that uses the company's money to invest in software startups through corporate venture capital investment funding
A004	A government agency that supports software startups by educating and funding those businesses in their initial phase through the agency's project, which aims to provide startups with the skills to develop their own business models
A005	A co-working space that provides support and facilities to enable startups and developers grow their businesses

Table 2 Software startup groups

Software startups groups	Number of startups
Tourisms technology	7
Financial technology	4
Lifestyle and personal service technology	4
Medical technology	3
Education and government technology	3
E-commerce and logistics technology	3
Industry 4.0 and clean technology	3
Property technology	1
Human resource technology	1
Agriculture and food technology	1
Total	30

4 Results

An analysis of the interview data reveals three conclusions based on the proposed research questions: (1) contextual factors of Thai startups, (2) support aspects, and (3) problems and future changes in Thailand affecting startups. Each section is detailed as follows:

4.1 Contextual Factors of Thai Startups

The analysis indicates subtopics, including tech startup definition, culture, and environment.

The Definition and Features of Software Startup Businesses Software startup businesses' features will be introduced and defined by comparing these businesses to small- and medium-sized enterprises (SMEs). Present-day software startups and SMEs have both received dual support from the government under various programs, including: the Small Industry Credit Guarantee Corporation, the Thai Credit Guarantee Corporation, and the Public-Private Collaboration of Support in SMEs, Startups, and Social Enterprises. However, current studies show that the features of software startup businesses and SMEs substantially differ in terms of how they use funds, both in terms of investments and business expansion; Table 3 provides further details provided by the supporter organizations.

Environment The environment that supports Thai software startups includes:

– *Co-working space*. The software startup collection center is a place where startup employees can gather and work. Moreover, it is also a place where these groups can share and exchange ideas. For example, startup companies that meet to work at co-working spaces can exchange ideas or ask for others' opinions by allowing others to test the applications that each startup develops. Additionally, these spaces are a source of new business partners, as these centers assemble groups of startups. Lectures and seminars are also held within these co-working spaces to educate startup employees on relevant knowledge; these seminars either include paid or free attendance.
– *Incubator center*. These aim to provide consultations for startup employees and connect their businesses, as these employees may be unaware of problems their companies may face during the initial software startup stage. Thus, incubator centers evaluate the startup's potential and make plans for the startup to conduct its business model. By comparing the problems that may arise within the startup company and its potential, the incubator center can later consider a startup company's future actions through an action plan. These accordingly guide the startup in its operations each year, and help the startup establish long-term 5-year company goals. Thailand's universities host incubator centers, although none has hired full-time software startup experts. Another incubator center can be found in organization A001 (Table 1); this incubator center has established for approximately 15 years. Moreover, the startups that join this agency will receive an expert business analysis regarding each startup's situation. A different incubator center will address each startup depending on their needs. However, Thailand currently has insufficient incubator centers to meet startups' demand, and no incubator center has received full government support. This contrasts Thailand's neighboring countries, such as Malaysia; their "magic" incubator center is a government agency that recruits around 100 startup teams each year to join the center for 6 months and learn according to the government's curriculum.
– *University*. The university is an important part in the Thai software startup ecosystem, as universities include experts in advanced software startups, such as artificial intelligence. Further, research is also brought into universities to establish real startups. Therefore, universities have an important role in producing startups that are technologically advanced and provide the country with startup research-related items.

Table 3 SMEs versus Software Startups in the Thai context

	SME	Software startup
Definition	A business whose products are purchased, then sold through a storefront. As SMEs have been registered, they are obliged to follow the entire business process	A newly emerged business that is expected to grow rapidly by combining the use of science, technology, and innovation. The business must grow exponentially and must be able to expand and replicate; each startup has a different business model with a clear business direction
Establishment	Although SMEs are established as new businesses, they still use general forms of business management	Each business is established to find its particular business model based on the product that is produced
Using technology	Some SMEs do not use technology	Startups use technology to assist in more rapidly expanding the business. This technology is used as a medium to communicate with customers through the business' applications and websites
Investment	Investments occur using the SME owner's own money, or they must find their own funds to invest, such as bank loans with one's property as collateral	Investments occur using the owner's own money or proposing ideas to investors in requesting investment funds. However, software startup businesses cannot receive loans from banks, because it is difficult to guarantee such loans
Growth	These SMEs grow organically by expanding to include multiple store locations. This requires the same capital needed to invest in the initial store, while doubling the amount of investment money	Exponential growth occurs without having to start from the beginning. Global growth is one way to expand, in that businesses grow from one country to another by increasing only 20–30% of their investment funds
Business expansion	These SMEs expand by opening more locations. The same business process occurs in expanding a store's branch from one to two stores, which in Thailand may simply involve an expansion from one province to another	The existing system can be further adjusted. For instance, a startup in one country can expand their business to other countries by merely converting the old system's language into the new country's language
Personnel	No more than 50 employees	A team or group of people operate startups from the same or different branches

Culture This factor impacts the software startup ecosystem. For example, some startups have no new ideas, but it is possible to bring in successful startups from other countries to Thailand, and refine them for use with the Thai people, who are also considered successful. In other cases, some startups are well known in other countries, but might not be marketable in Thailand due to cultural limitations and the differences in laws in each country. Therefore, although some startups may not be able to think of new business models, they can understand a particular country's culture and ways of life. Consequently, this kind of startup can also be successful in certain countries.

Our interviews indicated that the top 3 reasons for choosing Agile methods were as follows: (1) Agile development handles changes smoothly, even when introduced to projects late; (2) Agile development results in frequent feedback from the team, which helps to improve the quality of software by reducing project risk; and (3) Agile development enables software developers to develop software that meets customers' needs and requirements. For example, one interviewee said, *"Scrum helped the team to change the design when we have a new idea."* Four interviewees agreed, saying, *"Agile is a fast way to deliver, adapt to changes."*

Many startups mentioned Agile methods that they adopted to develop their software products or services. Although many startups had not previously used Agile methods (e.g., Scrum, eXtreme Programming), they tried to learn how to adapt these methods to their software projects. We found that 20 startups adopted Agile methods with their projects. The top choices of Agile methods implemented among Thai startups were Scrum (65%), eXtreme Programming (20%), Kanban (5%), and Lean (10%).

Besides Agile methods, all startups used some software engineering practices. Daily standup practices were adopted by 45% of the Thai startups. Prioritized backlogs, retrospectives, unit testing, short iterations, iteration planning, task board, and refactoring were adopted by 20%. Team-based estimation, coding standards, test-driven development (TDD), iteration reviews, continuous integration, pair programming, and single team (integrated development and testing) practices were adopted by 15%. Automated acceptance testing, release planning, continuous deployment, dedicated product ownership, open work area, story mapping, collective code ownership, and behavior-driven development (BDD) were adopted by 15%, while Agile game was adopted by only 5%.

4.2 Support Aspects

Currently, Thailand's software startups are supported by both public and private sectors. These software startups range from those in the idea-proposal stage to those that need investment funds to expand their businesses, as well as the various organizations that assist these startups; the details of each segment are as follows:

Private Sectors This involves companies that are not registered to trade on the stock market, and can be categorized as follows:

Venture Capital This mutual fund does not use their own company's funds to invest, but uses funds from multiple investors. The company later uses the money received from investors to invest in startups, in which the person investing must be able to accept the risk of investment. Moreover, the objective is to profit from the investment and return the profit received to those investors. For example, investors might provide a dollar to give a startup the investment money to grow, with the hope that the investor will receive 10 dollars back. Most investors would each have to invest millions of dollars, and the investor would have to follow up with the startup after investing by measuring the financial return to determine not only how much the startup has grown, but also the extent of returns it is making for its investors.

Corporate Venture Capital (CVC) This company uses its own funds to support the startup process without gathering money from other investors or opening up funds that receive outside funding. As the company invests its own money in startups, it aims to support startups associated with the company's operations. Moreover, the company may apply what the startup has created within the company. Further, this investment brings in new technologies to help enhance the business. However, the startups that joined the program with the leading company do not need to become the supporting company's subsidiaries. Startups that are supported by the company can create their own startups for other companies' use; these CVC-type companies will support startups greater than the Series A level, as those startups are already relatively stable.

Regarding startups' funding, for example, A002 will support startups in the financial technology field, as well as those related to financial technology. Moreover, the company can apply the startup's application to the company's queuing system. Other than supporting the startups directly associated with the company, the company can also support startups that are commonly useful in other parts of the business as well. As A002 holds many events, the company was forced to find a technology to assist in its event management; this application allowed eventgoers to purchase and pay for tickets.

As another example, A003's investments consisted of startups still in the idea-proposal and Series A phases. However, the company currently provides support only for startups of Series A level and greater, as this is less risky than supporting startups that aim to profit financially, and bringing in new startups to use within the company with new technology to enhance the business. Consequently, the chosen startups that the company prefers to invest in are those in Series A and above.

Crowdfunding This type of investment brings in investors and startups to meet, as if setting up a meeting between investors and startups. If the investor wishes to invest in startups, they can agree on how the former will make such an investment. Moreover, the crowdfunding company is responsible for finding startups that match the needs of the target investor group, and the investors can choose from these businesses. Crowdfunding occurs when startups are so overcrowded in certain countries, to the extent that investors cannot find their own startups.

Initial Coin Offering (ICO) This fundraising occurs in a form similar to crowdfunding, and the practice involves trading exchange rates into tokens for online trading. The startups that need funds from investors will write to explain what the startup will do for the business, including the business' type of startup, and to set a selling price. If investors are interested, they can buy the startup online with token coins, which approximates crowdfunding investments. However, ICO investments are more specialized in their investments than crowdfunding.

Matching Fund This is an investment partially funded by the government. If the startup wants to expand their business, the government will provide 50% of the funding for support, and startups can also buy their shares back from the government, if they prefer. For instance, a startup that wants to expand its businesses may need 10 million US dollars in investment money. The startup will then have to find investors to invest 5 million US dollars in the startup, while the government will fund the remaining 5 million US dollars. Therefore, when the startup is ready to operate, it can then buy back the treasury, which will also help decrease investment risks for those who had invested all their money in the startup.

Angel Investor This is a partial group of venture capital investors that uses their own funds to invest in startups currently working on various ideas. An agreement is made between the investors and the startup regarding the feedback the Angel investors will receive in return. Under this condition, the startup must provide some benefit to the Angel investor, which may be in the form of currency or stock; the startup can buy the stock back afterward.

Public Sectors The government supports technology startups through its investment. These investors from government organizations could include banks under the government's supervision. Further, the government also provides support through the investment funds earmarked for various startup projects.

This government project supports startups in various phases, from those that are still working on their ideas to bring them into an incubator center to those that want to expand their businesses by funding into the market.

Thailand's government has established a National Startup Committee composed of the deputy chief of the Ministry of Finance and committee members from 16 ministries the participants of which have planned to support startups for 5 years. Further, these members plan to recruit participants annually to join the project from 3000 startups currently in their idea stage.

The goal is to provide startups in the idea stage the support they need until they reach at least 300 businesses (10%); if any startups need marketing, they can then ask for financial support from the government. The government will provide financial support for 75% of the project, or up to 25,000 dollars. Within 1 year, the government's market was composed of provided funding for 80 startups annually. Before funding all 80 startups, a list was gathered of startups that requested supporting funds to allow the committee to analyze them. The committee scored the startups to determine which can characteristically and effectively garner funds and use them to their maximum advantage.

The government plans to support startups from their idea stage to the startup expansion stage. These projects can be divided as follows:

Startup Club Project The Startup Club Project is a collaborative project that involves brainstorming ideas from the advisor of the ICT Ministry, the Deputy Permanent Secretary of the Ministry of Finance, university professors, and groups of people establishing startups in Thailand. The collaboration in establishing this project reveals various problems in creating startups in Thailand; specifically, these employees lack a university education. Universities can teach basic academic courses, but their students still lack vocational skills. Therefore, the Startup Club Program, the participants of which are university students, was jointly designed to collect those with startup experience to help teach in universities.

Taokaenoi Technology Project This government project accepts startup companies that have only one idea. Basic selections are made based on the startup's concept, as startups with possible ideas are selected to join the program and further develop their business models. When a startup finally develops a decent business model, the program will then send the startup to other government programs—such as the National Science and Technology Development Agency (NSTDA) and National Innovation Agency (NIA)—to let the startup further develop their business model into actual products.

Coupons of NIA Innovation This government project accepts startups, and wants to develop their prototype products. Participating startups will be considered by judging the startup's concept based on the business model, which indicates whether the concept can be further developed. The project supports the creation of a prototype that could be transformed into a real product and can be used in an actual business. The budget in creating a product prototype is approximately 31,000–80,000 dollars per startup. Moreover, the startup must be able to create an actual product from the prototype. If the startup can do so, it can ask for financial support after the program ends from the NSTDA's startup voucher program to raise the funds to publicize their startup in the market.

Startup Voucher This government program accepts startups that must be publicized in the market. The startups that join this program must have a product that can actually be used and a solid customer base, and must already have both sales and earnings. The program was established to help accelerate startups' growth rate by marketing through various channels. For startups to participate in the program, there is a condition in receiving financial support, in that the program support is 75% of the startup's necessary funding, or does not exceed 25,000 dollars. Moreover, startups must pay the 75% of money that it needs the program to support first and bring the payment receipt to receive funds back from the program afterward. The program's selection process will be judged from the startup's qualifications, as the startup must have been established for no more than 7 years, has registered capital, and must only be technology related. After passing the initial selection process, the startup has to present their products to the committee. The committee will then judge the startup using selection criteria from the A001 agency, which includes an analysis of the startup's investment market and business management; the potential to establish an actual business will also be considered.

> Thai organizations that support and encourage software startups are government agencies, trade associations, or accelerators. Government agencies play an important role in providing support mostly to startups.

4.3 Problems and Future Changes in Thailand Affecting Startups

Our interviews reveal the respondents' opinions on software startups in Thailand, which can be divided into the following subtopics: (1) Groups of people who have established startups in Thailand, (2) The startup company's objective in joining the organization's project that supports software startups, (3) The problems encountered in Thailand's software startups, and (4) The future of technology startup businesses. Each subtopic is noted in detail below.

Groups of People Who Have Established Startups in Thailand The groups of people establishing software startups range from groups of high school students to those at retirement age. We found that regarding the groups of people establishing startups, students interested in this process are university students in the fields of science, information technology, and engineering. Startup groups composed of university students include groups with new, innovative ideas, although most of them cannot create their own business models. Moreover, they will encounter problems when their startup expands, as they cannot manage and arrange their organizations. Regarding startup groups composed of people with work experience, they can think of a business model and gain a deeper understanding in these startups; they can also manage their work better than groups of university students.

> Thai startup evolves in three ways. First, startups that fully develop are no longer considered startups, and have been incorporated into the stock market. Second, other companies can acquire or purchase startups to use them in the parent company or to develop the startup as a subsidiary. Third, the startup may have a lower growth rate than usual, or its growth may stabilize.

The Startup Company's Aim in Joining the Organization's Project that Supports Software Startups The startups that participate in the project have an objective to seek advice from experts in startups that are developing, and to also find networks in the startup's ecosystem. Moreover, and on the one hand, they aim to seek a customer base for the startup from the various companies that have joined the program. On the other hand, some startups have joined the program to find sources of financial support. However, it was found that the objectives in finding financial support are the least valued among the startups that participate in the program.

Problems Encountered in Thai Software Startups We found that, of the problems encountered in software startups and in supporting software startups in Thailand, a conflicting understanding exists regarding software startups' features. It was also found that some business operators want to join the software startup program to ask the government for financial support and further develop their applications or websites within their businesses. For example, a clothes-selling business is requesting financial support from the program to establish their own clothing website to sell clothes online. Another problem technology startups encounter is the scarce technology and innovation in Thailand, which makes companies unable to withstand rapid market changes. Further, a lack of human resources who are startup developers, such as software developers, are scarce. In particular, some startups reported that they lacked sufficient knowledge of software testing. One participant said, "There was a lack of knowledge regarding how to test the software, especially good testing was time-consuming." Another participant agreed, stating, "Unfortunately testing techniques tend to be a set of software engineering skills and practices that is often absent." Unfortunately, they indicated that they could not hire experienced engineers due to budget constraints.

The Future of Software Startup Businesses The future of software startup businesses in terms of new software startup development demonstrates that such businesses should be able to meet the needs of users in Thailand. The need to support software startups can be divided into the following groups:

1. Developing software startup. Currently, startup trends in Thailand still involve the development of startups for smartphone applications, and thus, startups in the form of applications will decrease the startup's sustainability.

   Advice has been offered from groups of investors and supporters, in that startups should develop to use advanced technology. Further, these groups have recommended suitable types of software startups that may suit the needs of the Thai people: a startup group in the tourism technology field. As Thailand is a land of tourism, any startup that develops according to tourists' needs may as well help to promote tourism. For instance, an application could be created that responds to the needs of Chinese tourists in Thailand.

   Currently, tourism technology startups merely involve e-commerce, such as applications that sell tours. However, startups have yet to be found that enhance Thai tourism. The second necessary startup group involves medical innovations and public health (medical technology), and the last group involves agricultural and food technology. Agricultural startups are rarely seen, although agriculture is an important practice in Thailand and large companies control the market, making agricultural startups rare. The interviews have led to various recommendations, such as how to handle one square meter of land to be able to cultivate good crops.

2. Educating those who want to establish startups by opening an incubator. This is due to insufficient incubators and the requirement that incubators must be monitored by experts in consultations, and forcing startups to properly prepare themselves for their public presentation. Further, advice and recommendations are provided regarding technology promotion as well as knowledge in different aspects of the business. Moreover, the public sector must promote startups to easily register and help them by making laws that support them. For example, laws for investors can compel them to invest easier and implement more incorporated taxes paid during investments. Further, changes can be made, such as the government in the present day will exempt investors from taxes who only invest in domestic startups only. Changes should be implemented regarding tax exemptions for investors in domestic startups, while other foreign investments are taxed normally. Moreover, the need exists for the government to support human resources among software development, investments, and management knowledge. Finally, startups' knowledge is necessary; if the startup expands, for example, the startup must have additional knowledge in managing its people.
3. Funding and partners. According to the study, 90% of interviewed startups said that their most pressing concern was finding funding sources. Approximately 80% of startups said they want to collaborate with big companies because big businesses have more money to spend on startups. The study also found that startups require a long-term partner instead of short-term business deals. Approximately 90% of startups indicated that they look for long-term strategic partners.

5 Discussion and Recommendations

Results from the latter part of the study indicate support for software startup development from Thailand's public and private sectors, which provide support in the form of investors and incubator centers. It was also discovered that demands were made in opening the incubator center as a source for newly launched startups to seek counsel and a source of knowledge in different areas of the startup. These include: giving advice on the initial stage in establishing software startups, providing counsel regarding the steps to find investors, and educating startups about how to manage their businesses when their startup is fully grown.

By comparing the startup ecosystem in Thailand to the ecosystems studied by previous works [5, 12], we found that some elements differ. Figure 1 illustrates Thailand's software startup ecosystems, as well as previous work. The gray box presents the elements that only exist in the Thai context. The blue box denotes the element as existing in both Thai and previous work. The white box presents the element that existed in previous work, but that does not exist in the Thai context.

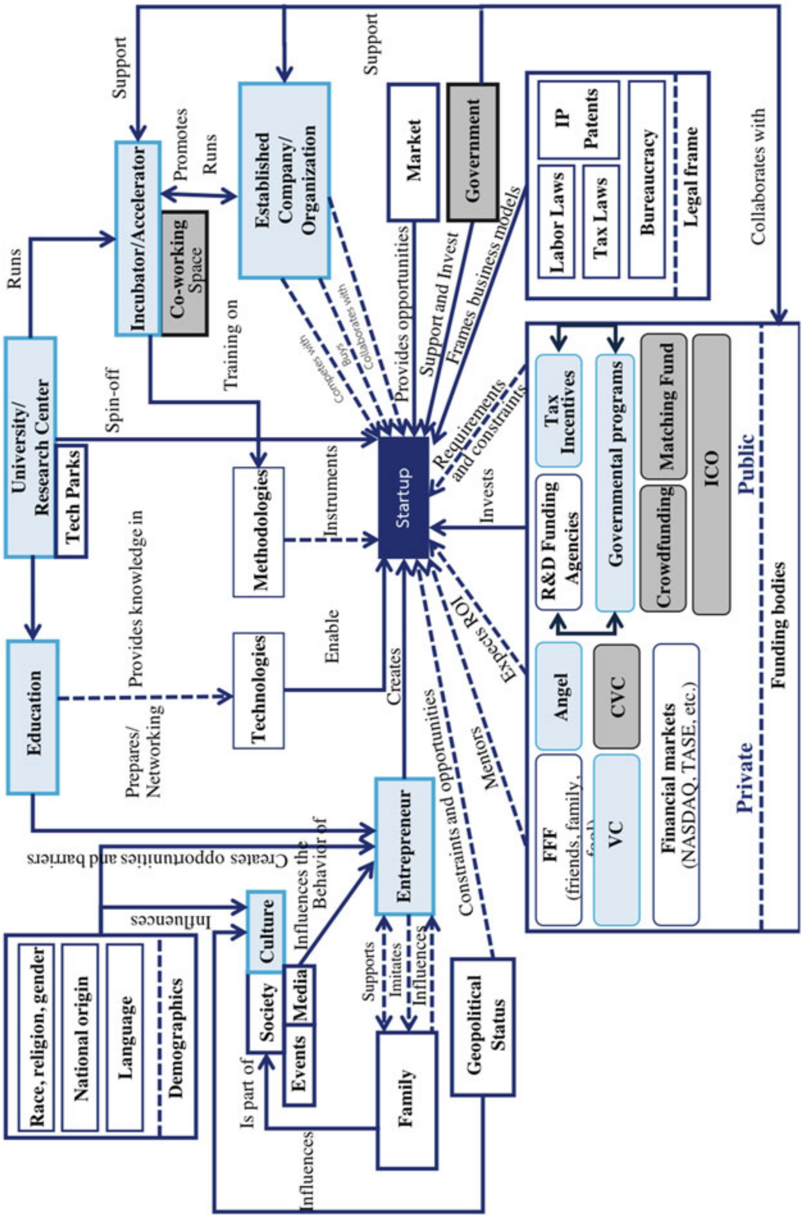

Fig. 1 Technology startup ecosystem in Thailand and previous work

This study of software startups in Thailand reveals issues that can be summarized to provide recommendations for those involved in software startups, as follows:

1. The study reveals the difficulty in understanding software startups' significance, which indicates conflicting aspects of the features of software startups. Therefore, the scope of software startups should be clearly defined to provide the same understanding for those who want to work or invest in software startups.
2. Regarding software startups' development, the study demonstrates that the startups of interest that meet the needs of users in Thailand are software startups in the tourism technology, medical technology, agricultural, and food technology genres, as well as software startups that use advanced technology in which external research can be applied in developing new software startups.
3. Regarding those who influence the internal software startup ecosystem, this study discovered that universities cannot organize their courses to guide students in developing their own software startups. Consequently, a group of students who establish startup businesses do not have the expertise to consider a business model. However, the university plays an important role in developing technologically advanced startups, and startups that bring in external research to apply in creating software startups.
4. The study reveals a lack of resources in developing software startups, or specifically, such human resources as those with software development knowledge and business and management expertise. This can cause the startup to lose growth opportunities, and therefore, knowledge should be provided in terms of preparation and the importance of personnel within the software startup organization.
5. Supporting software startups reveals the different aspects of support between the public and private sectors in the incubator center project for software startup producers. The first issue is the project participants' thought process toward a business model, as public sector projects will bring in actual business models for the group of newly established startups. In contrast, the private sector's project allows startup initiators to think of their own business models, and the project's staff will find groups of people who are actual users to talk with the startup group. Investors in Thailand are mostly venture capital or corporate venture capital (CVC) companies. Angel investors have smaller amounts of capital, possibly because Thai business owners do not yet understand this business model. Some venture capital businesses will not invest during the early phases in case the startup does not succeed. However, many big companies established CVCs in the year 2017. This rise in large company investments could be due to increased confidence in the enterprise potential of startups in Thailand. Companies that established CVCs during the past year came from various types of business groups, including AddVentures, Digital Ventures, Beacon, Bualuang Ventures, Krunsri Finnovate, Invent, Ascend Capital, Central Online, Benchachinda, PTT, J Ventures, Dusit Thani, SCAble, Ananda, Siri Ventures, MFEC, Fuchsia14, and more.

6. The study has demonstrated that people who establish software startups faced different problems in terms of technology, the knowledge in creating startups, asking for support funds, and in terms of management when startups are fully grown. Therefore, an incubator center that specializes in software startups should be open to acting as a main source not only for those who are interested in creating startups, but also for startups that need more information to expand their business. This will also serve as a main source for linking software startup ecosystems together to transmit news among these entities. Moreover, software startups' technological development could be standardized to promote movement in the same direction. Additionally, the study reveals that people who establish software startups want to participate in startup supporters' projects; ultimately, people can find networks and connections with those familiar with the topics that are of interest to startups. A customer base among the startup supporters can also be ascertained.

Based on software startup recommendations, we summarized the type of support that the Thai government should provide:

– The government must set up special state agency to support startups, allowing them easier to do business.
– The government must provide the marketing channel for foreign markets, such as business-to-government (B2G) and business-to-business (B2B).
– The government must groom new talents by providing financial incentives to new startups.
– The government needs to prescribe more and better policies and laws conducive to technology and innovation.

In addition to government assistance, we recommend that other success factors be adopted by Thai startups, including:

– A follow-up to the software startup in each stage
– The startup's internal structure
– The organization's internal regulations and policy arrangements
– The startup's economic background

6 Conclusion

This research involved interviews with those involved in software startups, from the perspective of software startups and investors. This is one approach to help foster the software startup ecosystem in Thailand. Further, this may compel the software startup ecosystem to adjust to better meet the needs of the software startup's organizer, investors, and those involved in other parts of the business. This research will also help increase the empirical evidence that benefits those software startup researchers interested in studying software startup-related topics.

In the future, this research can guide educational studies in applying software engineering methods in developing software startups. This can be accomplished using an empirical software engineering method that consists of field research. A sample group of software startup developers can be surveyed using questionnaires to collect data, which can be used to analyze the startups' quantity and quality. These surveys can also be used to analyze the results in using software engineering methods in developing software startups.

References

1. Abrahamsson, P., Nguyen-duc, A., Baltes, G.H., Conboy, K., Dennehy, D., Sweetman, R., Edison, H., Shahid, S., Wang, X., Garbajosa, J., Gorschek, T., Unterkalmsteiner, M., Hokkanen, L., Lunesu, I., Marchesi, M., Morgan, L., Selig, C., Oivo, M., Shah, S., Kon, F.: Software startups–a research agenda. e-Informatica Softw. Eng. J. **10**(1), 1–28 (2016)
2. Berg, V., Birkeland, J., Nguyen-Duc, A., Pappas, I.O., Jaccheri, L.: Software startup engineering: a systematic mapping study. J. Sys. Softw. **144**, 255–274 (2018)
3. Coleman, G., O'Connor, R.V.: An investigation into software development process formation in software start-ups. J. Enterp. Inf. Manag. **21**(6), 633–648 (2008)
4. Corporation, M.D.E.: MDEC (2019). https://mdec.my/
5. Cukier, D., Kon, F., Krueger, N.O.: Designing a maturity model for software startup ecosystems. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 9459, pp. 600–606. Springer, Cham (2015)
6. Daniel Chan, W.L.: China updates high and new technology enterprise tax rules: 5 key implications (2019). https://www.dlapiper.com/en/us/insights/publications/2016/02/china-updates-high-and-new-technology
7. Duc, A.N., Khalid, K., Lønnestad, T., Bajwa, S.S., Wang, X., Abrahamsson, P.: How do startups develop internet-of-things systems: a multiple exploratory case study. In: Proceedings of the International Conference on Software and System Processes. pp. 74–83. IEEE, Piscataway (2019)
8. Getlinks: STARTUP THAILAND JOB FEST 2017 (2018). https://getlinks.co/startupthailandjobfest
9. Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: the greenfield startup model. IEEE Trans. Softw. Eng. **42**(6), 585–604 (2016)
10. Kajko-Mattsson, M., Nikitina, N.: From knowing nothing to knowing a little: experiences gained from process improvement in a start-up company. In: Proceedings International Conference on Computer Science and Software Engineering, CSSE 2008. vol. 2, pp. 617–621 (2008)
11. Klotins, E., Unterkalmsteiner, M., Gorschek, T.: Software engineering in start-up companies: an exploratory study of 88 startups. Empir. Softw. Eng. **24**(1), 68–102 (2019)
12. Kon, F., Cukier, D., Melo, C., Hazzan, O., Yuklea, H.: A conceptual framework for software startup ecosystems: the case of Israel. In: Department of Computer Science, University of São Paulo, Technical Report RT-MAC-2015-01 (2015)
13. Network, G.R.: Software Startups Global Research Network (2018). https://softwarestartups.org/
14. Ping, C.K.: New one-stop financial technology office for startups launched (2019). https://www.straitstimes.com/business/economy/new-one-stop-financial-technology-office-for-startups-launched

15. Seaman, C.: Qualitative methods in empirical studies of software engineering. IEEE Trans. Softw. Eng. **25**(4), 557–572 (1999)
16. Startupsg.net: Startup sg equity (2019). https://www.startupsg.net
17. Techsauce: Thailand Tech Startup Ecosystem Q1 2017 (2018). https://techsauce.co/report/thailand-tech-startup-ecosystem-q1-2017
18. Thai Venture Capital Association: Thai Startup Founder Survey 2016 (2018). https://brandinside.asia/thai-startup-founders-survey-2016/
19. Xu, L.: Why you might build your start-up in China over silicon valley (2019). https://hackernoon.com/why-you-might-build-your-start-up-in-china-over-silicon-valley-b23ed7d63951

Part IV
Software Startup Education

Software Startup Education: A Transition from Theory to Practice

Rafael Chanin, Afonso Sales, and Rafael Prikladnicki

Abstract New software startups are born every day around the globe. Dropbox and Netflix are examples of successful startups. However, failure is the fate of most of them. Several facts, such as market competition or lack of resources, can impact the destiny of a startup. Nonetheless, little has been explored in terms of the impact of software startup education on the success or failure of startups. Even though universities are adapting their curriculum in order to embrace such important subject, the challenge relies on how to provide real-world experiences for students to develop relevant startups. Hence, this chapter intends to present the main contributions, initiatives, and lessons learned found in the literature regarding software startup education.

Keywords Software startup education · Software startup · Entrepreneurship

1 Introduction

In past years, we have observed amazing advances in technology. The Internet is probably the most remarkable of them. Nowadays, anyone can learn software development and create systems that can impact the lives of millions of people. Facebook and Dropbox, to name a few, are examples of such endeavors. When these organizations are in their first years of life, they are called *startups* [4]. Most startups follow the *lean startup* methodology [55], which combines short software development cycles with constant interaction with potential users/customers. By learning from these interactions, a startup can reduce its risks [13].

Startups run against the clock; if they do not find a sustainable business model before running out of resources, they will fail. In this scenario, startups must find how to make money. This is done by running experiments until a business model is tested and found [17]. Unfortunately, most startups do not succeed [29]. Although

R. Chanin (✉) · A. Sales · R. Prikladnicki
School of Technology, PUCRS, Porto Alegre, Brazil
e-mail: rafael.chanin@pucrs.br; afonso.sales@pucrs.br; rafaelp@pucrs.br

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_13

there are many factors that could lead to the failure of a startup, bad software engineering practices is known as being a key reason [16, 29, 36].

Several publications concerning software development processes in the context of a startup can be found in the most important database sources [27–29, 43, 47]. However, there are not so many studies focused on how software startup processes are taught to students in an academic environment.

> Companies and the academic community started to pay attention to startup-related content. However, studies on how to deliver this content in the classroom are still scarce.

Nonetheless, there are studies describing how computer-related courses and programs have included entrepreneurship content into their contexts [19, 31, 41, 51]. One of the biggest challenges reported is the lack of a realistic environment for students to work on these type of projects [52]. Since the main goal of a startup is to solve real-world problems, faculty must find strategies to provide the right environment to students.

In addition, these studies also pointed out that being technically competent is really important, but it is not enough. Knowing how to develop, market, and sell products and services is crucial to become a real entrepreneur. Several institutions are already providing programs and courses focused on entrepreneurship in order to fulfill this need [11].

In this context, the goal of this chapter is to present the main contributions found in the literature regarding software startup education. This information was gathered by running a systematic mapping process on this content. This work is an extension of a preview paper already published in the scientific community [12]. Since this topic is new and it is growing, we understood that it would make sense to update the information.

2 Research Method

In order to collect the aforementioned information, we carried out a systematic mapping following the recommendation of the most influential researchers in this area [8, 39, 50]. Figure 1 presents the process undertaken to perform the systematic mapping study. The remainder of this section depicts the planning of each step of this study.

Studies were identified by running the search string presented in Table 1. This process was performed by following the guidelines proposed by Kitchenham and Charters [38].

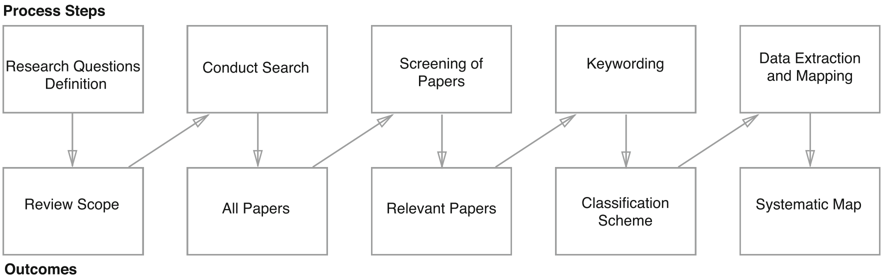

Fig. 1 Systematic mapping process (adapted from Petersen et al. [50])

Table 1 Search string

Population	(Software Engineering OR Software Development)
	AND
Intervention	(Software Startup OR Startup OR Entrepreneurship)
	AND
Outcome	(Education OR Undergraduate OR Graduate OR
	Teaching OR Educating OR Training)

Table 2 Search strategy

Databases searched	ACM Digital Library
	IEEExplore
	Scopus
	El Compendex
	Science@Direct
Selection Criteria	Available online
	Written in English
	From 1998 to May 2018
	In: Journals/Conferences/Workshops/Symposiums
	4 pages minimum
Search applied to	Title
	Abstract
	Keywords

The search strategy is depicted in Table 2. The database sources were chosen based on the list proposed by Kitchenham and Charters [38]. Two databases (Citeseer library and Inspec) were left out of this research due to technical difficulties in using these platforms. In regard to the publication period, we decided to begin in 1998 since this is the time in which the concept of *software startup*, as defined by Ries [55], started to be formed and studied.

Moreover, we only included papers that were accepted in journals, conferences, workshops, and symposiums. Extended abstracts, keynote presentations, and papers

Table 3 Retrieved papers

Database	Papers
ACM Digital Library (http://dl.acm.org/)	58
IEEExplore (http://ieeexplore.ieee.org/)	56
Scopus (https://www.scopus.com/)	67
El Compendex (https://www.engineeringvillage.com/)	85
Science@Direct (http://www.sciencedirect.com/)	2
Total	268

Table 4 Studies' basic information

Information retrieved	Explanation
Database	ACM, Science@Direct, Scopus, IEEExplore, El Compendex
Title	Paper title
Year	Year published
Authors	List of all authors
Type of forum	Journal, conference, workshop, symposium
Abstract	Paper abstract
Keywords	Paper keywords
Status 1	Duplicate
Status 2	Do not fit into criteria
Status 3	Is relevant

with less than 4 pages were also excluded from the research criteria since they usually do not present in-depth analysis. The focus was to identify papers that could present at least some preliminary studies on the topic. After running this process, we came across 268 papers. Table 3 presents the number of publication retrieved from each database.

Once papers were retrieved, a spreadsheet was created in order to organize the information for the screening process. Table 4 summarizes the information gathered from each selected study.

The screening process started by excluding duplicates, which accounted for 76 studies, leaving the spreadsheet with 192 papers. After that, we followed the exclusion criteria (as defined in Table 2), leaving the spreadsheet with 115 papers. Finally, in the last step of the screening process, we read the title, abstract, and keywords in order to verify whether the paper is relevant in regard to our research goal. At the end, our screening process led to 39 primary studies to be fully analyzed.

The key-wording process, which is illustrated in Fig. 2, was done following the guidelines from Petersen et al. [50]. It starts by reading abstracts in order to look for keywords that identify the main contributions of the paper. The goal is to create a set of categories in which papers can be grouped. If meaningful keywords cannot be found by reading the abstracts, researchers may look for them in the introduction and conclusion sections of the papers. It is worth mentioning that the classification scheme can evolve and change during the systematic mapping

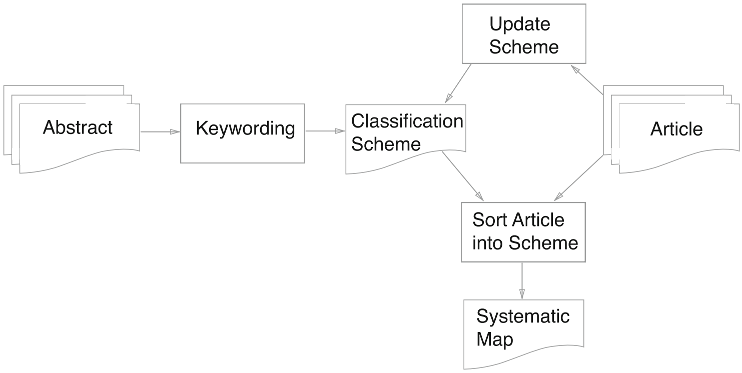

Fig. 2 Classification scheme workflow (adapted from Petersen et al. [50])

process. As researchers read the papers thoroughly, new categories/classifications may appear and others might merge or disappear.

In addition to the information mentioned in Table 4, we added the following information to the spreadsheet:

- Focus Facet: the categories created during the classification scheme process
- Contribution Facet: type of contribution. Based on a work from Shaw [63], and from Paternoster et al. [47]
- Research Method: method used on the research (case study, survey, etc.)
- Research Type: type of research (adapted from Wieringa et al. [66])
- Paper Quality: a grade from 0 to 10, based on the work from Salleh et al. [59]

Table 5 presents the systematic map overview after performing all steps aforementioned. The codes for our classification are depicted as follows:

- Research Method:
  - Experiment (EX);
  - Empirical Study (ES);
  - Case Study (CS);
  - Survey (SU).
- Research Type:
  - Experience Paper (EP);
  - Philosophical Paper (PP);
  - Evaluation Research (ER);
  - Validation Research (VR);
  - Solution Proposal (SP);
  - Opinion Paper (OP).

Table 5 Systematic map overview

1st author (year)	Research method	Research type	Contribution	Focus	Paper quality
Fagerholm (2017) [24]	EX	EP	LL	TH	9.5
Génova (2016) [26]	ES	PP	ML	TH	9.5
Järvi (2015) [34]	CS	ER	LL	TH	9.5
Schilling (2010) [61]	CS	ER	GL	TH	9.5
Adorjan (2017) [1]	SU	ER	LL	TH	9.0
Buffardi (2017) [9]	EX	ER	GL	MD	9.0
Izurieta (2016) [33]	CS	EP	FM	RP	9.0
Zaina (2015) [67]	CS	ER	FM	TH	9.0
Chesney (2014) [15]	CS	ER	GL	RP	9.0
Currie (2011) [18]	CS	EP	FM	RP	9.0
Salas (2017) [58]	SU	EP	LL	TH	8.5
de Lange (2016) [21]	CS	VR	FM	EV	8.5
Zhang (2015) [68]	EX	SP	ML	TH	8.5
Daimi (2008) [20]	ES	SP	GL	TH	8.5
Joseph (2006) [35]	CS	ER	LL	MD	8.5
Boutell (2017) [5]	SU	ER	LL	RP	8.0
Devadiga (2017) [22]	SU	EP	GL	TH	8.0
Ribeiro (2016) [54]	CS	EP	LL	EV	8.0
Vitolo (2016) [65]	EX	ER	LL	MD	8.0
McMahon (2014) [42]	CS	EP	LL	TH	8.0
Kaltenecker (2013) [37]	ES	PP	LL	TH	8.0
Chenoweth (2008) [14]	SU	EP	GL	TH	8.0
Buffardi (2017) [10]	CS	SP	FM	MD	7.5
Porter (2015) [51]	CS	SP	LL	MD	7.5
Nguyen-Duc (2016) [44]	CS	ER	ML	TH	7.5
Bharadwaj (2014) [3]	EX	EP	FM	TH	7.5
Breytenbach (2013) [6]	CS	EP	LL	EV	7.5
Ko (2017) [40]	ES	EP	AI	EV	6.5
Heintz(2014) [32]	CS	EP	LL	TH	6.5
Pauca (2012) [48]	CS	SP	FM	RP	6.5
Ford (2004) [25]	CS	EP	LL	MD	6.5
Barbe (2010) [2]	ES	EP	ML	EV	6.5
Gross (2000) [30]	SU	ER	LL	TH	6.5
Q.-Sarmiento (2018) [53]	CS	SP	ML	TH	6.0
Sun (2009) [64]	ES	PP	ML	EV	5.0
Sarraipa (2016) [60]	CS	SP	AI	EV	4.5
Engelsma (2014) [23]	CS	EP	LL	RP	4.5
Rioja Del Rio (2014) [56]	ES	OP	TL	TH	4.5
Pauli (2008) [49]	CS	EP	LL	RP	2.0

- Contribution:
  - Lessons Learned (LL);
  - Model (ML);
  - Guidelines (GL);
  - Framework/Method (FM);
  - Advice/Implication (AI);
  - Tool (TL).
- Focus:
  - Teaching (TH);
  - Multidiscipline (MD);
  - Real Projects (RP);
  - Environment (EV).

2.1 Threats to Validity

There are several threats that could invalidate the systematic mapping. If the search strategy is not performed correctly, the retrieved papers may not account for all studies that could answer the research questions. Moreover, data extraction and paper classification are also important steps that should be done carefully by all researchers involved. Finally, it is important to pay attention to researchers' bias. In order to mitigate these threats, we followed the recommendations from Petersen et al. [50] during the whole systematic mapping process.

Aside from the systematic mapping process itself, we understand there are two important threats that require attention. The first one is related to the publication bias. We are more likely to find papers reporting positive experiments regarding software startup education rather than failure ones. It is very difficult to mitigate this risk since we only have access to published data, naturally. The second threat, and most important one, has to do with the interdisciplinary aspect of software startups. Since this topic overlaps with entrepreneurial education, this work could be missing relevant sources from the business area. Even though we are aware of this issue, we decided to focus only on computer-related sources. As future work, we understand it is worth developing a cross-area research in order to verify whether new insights may arise.

Another important point worth mentioning is the rationale behind choosing to run a systematic mapping review rather than a systematic literature review. We chose the former because software startup education has just started to be explored by the scientific community. Therefore, we wanted to draw an overview of this topic in order to identify the main contributions that have been made.

3 Results

We divided our results into two parts. The first one is related to tools, models, methodologies, and frameworks applied in a software startup education context. The second one depicts best practices found in the literature.

3.1 Tools, Models, Methodologies, and Frameworks

Regarding methods and methodologies, Zaina and Álvaro [67] proposal combines lean startup [55] user-centered design [57]. The idea is to foster entrepreneurial and innovation behavior in a software engineering course. The authors claim that computer-related courses focus mostly on technical issues, such as programming, setting aside important soft skills, such as creativity, problem-solving, and conflict resolution. In this study, two case studies were conducted in order to verify the effectiveness of the proposed methodology. According to the authors, bringing lean startup and user-centered design into the classroom stimulate students into learning important business concepts. Perhaps the most relevant one is the idea of keeping users in the center of the process, making students understand their real problems and needs.

Main areas of contribution:

✓ Business Model Canvas
✓ Customer Development
✓ Design Thinking
✓ Agile Methodology

In another study, Buffardi et al. [10] claims that it is very difficult to emulate real-world situations in an academic context. Working with "toy" projects is interesting and good when learning technical skills, but it does not bring real challenges that a real project presents. For instance, it is hard to emulate market competition and customer relationships. Even if instructors create scenarios to challenge students, there are several limitations. In addition, this study brings interesting insights (taken from Nurkkala and Brandle [45]) regarding the gaps between industrial software engineers' and software engineering students' experiences. They are sixfold:

1. Real product versus a project
2. Long duration versus short duration
3. Low turnover versus high turnover
4. High complexity versus low complexity
5. Needs maintenance versus no maintenance
6. Real customers versus no customers

Hence, Buffardi et al. [10] proposed a methodology in an attempt to minimize these issues. The idea is to create a collaboration environment that connects software engineering and entrepreneurship students. The latter would act as customers. Although engineering students have reported that the experience was interesting and relevant to them, this process still mimics a real-world context. Even though the authors understand this is not ideal, at least it gives students a taste about the process of creating and developing a startup. It important to point out that instructors need to analyze the trade-offs. Depending on the context in which the course is offered, it may not be possible to actually work with real-world projects.

Pauca and Guy [48] also highlight the difference between working in an academic context versus developing a real-world project. According to the authors, most of the software projects developed in a classroom are uninteresting, trivial, or not meaningful to students. Engagement and excitement increase only when students work with real problems and challenges that can affect society positively. This concept was labeled *socially relevant computing* [7]. In this sense, the methodology proposed by the authors consists of combining software development, agile methods [62], and social relevant projects. Results from a case study indicate that this approach stimulates the development of startups. Moreover, several students continued to work on their projects after the end of the course, proving they were really connected to problems they were trying to solve.

In regard to models, Génova and González [26] argue that there are three stages in a complete engineering education process:

1. Instruction: traditional education environment, with exams and projects
2. Training: when students receive a problem and choose the mean to solve it
3. Mentoring: when students can self-propose their own learning objectives

The authors claim that "*education is incomplete if the third stage is not reached*." However, there are some challenges that educational institutions must overcome in order to achieve the third stage. One of the most important ones is related to assessment. If students define their own learning objectives, it can be very hard for faculty to fairly evaluate students. Even though the authors understand there are several challenges in this process, they argue that if computer-related programs do not provide opportunities for students to achieve the third stage, they will not become real engineers; they will just be "programmed machines." From a startup perspective, it is crucial to achieve the third stage. Soft skills are a must when it comes to developing real businesses.

In another interesting study, Zhang [68] proposes a model that combines business, environment, and technology. The idea is that these three components play a key role in the process of teaching startup-related concepts. As already mentioned in this chapter, business and technology form the foundation for this learning process. However, it is important to point out that it can be challenging to deliver both contents in the same context. There is a need to coordinate efforts between two different schools/departments (business and technology), and this is not always easy. In regard to the environment component, Zhang argues that students might and should take advantage of resources and events that the university provides. Just to name a few, several universities offer incubation process, mentoring, connections, and networking events. By connecting these assets with the course, the whole process can be enhanced.

Barbe [2] proposes a model that intends to connect all aspects of a software startup development process. It starts from the basic technical knowledge, and goes all the way to business acceleration and funding. The rationale behind this approach is that even though startups can be created by technical founders, most of them lack business and soft skills. Hence, they will either fail or will need to hire people with the skills they do not have. Therefore, students not only learn the technical foundations for developing a startup, but they are also exposed to the business side of a software startup development process.

When it comes to tools, Rioja Del Rio et al. [56] suggest the use of the Business Model Canvas (BMC) [46] so students can oversee the big picture of the process. The idea is that the BMC helps students into analyzing all aspects of a given business model and not only the technical parts. The BMC entails the value proposition of the business, the customer segment, the channels to reach customers, the relationships established with customers, key resources, key activities, key partners, revenue streams, and cost structure.

As far as we are aware of, there is no single approach when it comes to models, tools, frameworks, and methodologies applied in software startup education. As we presented, several strategies have been applied in order to address software startup-related content. Some of them are focused on encouraging critical thinking, big-picture thinking, and creativity, while others focus on attention to detail, method, technical issues, and in-depth analysis. Since courses have limited resources and different focus, faculty need to evaluate the trade-offs associated with each approach.

We can summarize our findings in this section in the following four points:

- **Business Model Canvas**: very useful since it helps students create a vision for their business model. It is especially appropriate when teaching to technology students, since this tool goes beyond the technical product and also focuses on other aspects of the business model.
- **Customer Development Process**: this methodology helps students understand that the customer should be in the center of the process. By running experiments, students can take actionable steps in order to validate business hypothesis.
- **Design Thinking**: creativity is a key to a software startup development process. This approach helps students in developing creative experiments and solutions in order to get to a product or service that matters to users.
- **Agile Methodology**: when it comes to software development, agile is the preferred approach. This is no surprise since the software development process should be flexible and open to constant change due to the characteristics of a startup.

3.2 Best Practices in Software Startup Education

The best practices found in the literature were grouped and organized in the following four categories:

1. **Real Projects**: Faculty should avoid at all cost to work with "toy" projects. As already mentioned in this chapter, when this is the case, learning is limited to technical aspects, and only a few soft skills, such as teamwork and conflict resolution, are explored. Students should work on projects that matter to them and that are connected to real problems. Therefore, the best approach is to leave the floor open to students to define their projects. However, this is not always simple, since students might have a hard time finding a meaningful project to work on. In this situation, a good alternative is to connect students outside stakeholders from the industry in order to look for problems worth solving.
   When this happens, students not only gain the opportunity to connect themselves with corporate managers and executives, but it also helps them in finding real problems. One important remark is that faculty should never force students to pick a given problem to solve; if there is no excitement and engagement, it is very likely that students will not fully learn the process.
2. **Multidiscipline**: Whenever possible, faculty should aim at cross-discipline collaboration. For instance, technology and business students should participate in the same course, working together on their projects. Although this idea may require collaboration and coordination among faculty from different colleges, it is definitely a great opportunity for students to take different perspectives on projects and ideas. In this scenario, students probably learn more from one another than from the instructors.
   Naturally, faculty should pay attention to form groups with students with different backgrounds. One important point here is that this kind of situation may require patience and ability to solve conflicts by faculty members. The recommendation is to set the ground rules right at the beginning of the semester and reinforce them along the way. In addition, students should also develop their own guidelines in order to address decision-making and conflict resolution.
3. **Environment**: The connection with outside stakeholders also plays a key role in the software startup educational process. To begin with, we have the real customers and users that must be found by students. In this sense, faculty should create opportunities for students to connect to them. If students do not succeed in this process, faculty may look for partners, such as startup founders and executives, in order to help students in getting outside feedback. This gives students the opportunity to discuss their projects with more experienced entrepreneurs.
   Moreover, students should also take advantage of the university ecosystem. Connecting the course with networking events, hackathons, and even with incubators and accelerators gives students another perspective on the startup context. There are several cases reported in which students pitch their project ideas in real startup events.

4. **Teaching**: Several insights were found regarding teaching strategies. In regard to team formation, the ideal number suggested is four or five. Working with less than four members may lead to a poor team composition. On the other hand, big teams may require a lot of coordination, not to mention the chance of having students not getting completely involved in the projects.
   All teams should have a leader, who acts as team liaison. Aside from that, teams and instructors should meet on a regular basis in order to present the work being done as well as to receive general guidance on the projects. In addition, it is very important the teams present their work to one another several times during the semester. This approach not only helps students in improving and developing their presentation skills, but it is also a great opportunity to collect feedback.
   Regarding assessment and course evaluation, several authors argue that startup courses should not have final exams without the reflection of explicit learning goals. Since "anything can happen" during the development of a startup, the learning happens throughout the process, especially when students are running experiments. For instance, when students talk to potential customers, they may learn a lot about the problem they are trying to solve. Therefore, assessment is done by analyzing students' project progress, as well as by asking for team and personal reports. In this sense, students should get used to documenting every step they take, from ideation to the final presentation. Since this assessment process is not usual for students, it is important for instructors to set the ground rules and explain the assessment process very clearly in the beginning of the semester.
   As one can imagine, traditional lectures are not the best approach for this type of content; a flipped classroom is ideal in this scenario. The process of developing a startup is way more important than the end result. Therefore, instructors should focus on giving students the opportunity to experience the ups and downs of startup. Focusing only on a final deliverable or in an exam does not make sense in this context.
   When it comes to software development processes and tools, the authors claim that students should use the same programming language. The idea is that it becomes easier for instructors and students to help each other when a common language is defined. The downside of this approach is that the defined language/technology may not be the ideal one for a given project. In this sense, faculty should analyze the trade-offs of going with a single programming language.
   Finally, it is important to point out that product development is different from innovation development. The former follows a straightforward path, while the latter is chaotic and unpredictable. In order to add structure to the innovation process, it is important to add tools and methodologies, such as the *Business Model Canvas* and the *Customer Development* process.

As already pointed out in this chapter, when it comes to software startup education, it is not easy to provide a realistic setting for students. It usually comes at the expense of processes, practices, and goals. Even when connections with real-world problems are possible, in most cases students do not keep working on their project after the end of the semester. Nonetheless, there are several interesting cases reported with great insights and lessons learned. For instance, when faculty is able to connect the course with the university ecosystem, especially incubators, there is a higher chance for projects to actually become real startups.

"Greenfield" areas for improving software startup education

✓ Real Projects
✓ Multidiscipline
✓ Environment
✓ Teaching Methodology

A final note worth mentioning is related to courses ordering and organization. Heintz and Klein [32] suggest that computer-related programs should begin by showing students the "big picture" rather than going straight to fundamental courses, such as math. The idea behind it is that students do not get engaged if they do not understand the purpose of a given content. By the time students oversee the whole process, there is a higher chance they will understand the value of certain contents.

4 Conclusion

In this chapter we presented the main contributions on software startups education found in the literature by running a systematic mapping review. The takeaway lessons from these studies were divided into two main points:

1. Tools, models, frameworks, and methodologies applied in software startup education
2. Best practices in software startup education

In regard to the first point, we concluded that there is no unique approach applied in software startup education. It seems that several variables account for this matter. For instance, if the course focuses more on business modeling, instructor tends to get deeper on big-picture thinking and creativity. On the other hand, when the course focuses on software development, technical issues arise the most. We believe that regardless of the course approach, there is room for a more consolidated approach when it comes to software startup education. Students should be exposed to all aspects of a startup lifecycle; from ideation and business modeling, to software development and customer interactions.

Our second point, which was divided into four categories—real projects, multidiscipline, environment, and teaching—brought us insights that are consistent with the information found in the literature about software startups [27, 29, 43, 47]: the goal of a startup is to solve a real-world problem. This is done by combining a multidisciplinary team and the right environment. In the educational context, however, addressing real-world problems is still a challenge. Moreover, it is not always possible to have students with different backgrounds in the same course.

It is important to point out that this work, which is an extension of a preview paper already published [12], did not bring new tools, methodologies, or practices. However, we could observe that not only the academia, but also the scientific community is paying more attention to this topic. Although research in this area is still in its first steps, we understand that there is a lot of opportunities to improve the experience for students when learning about software startups.

Acknowledgements This work is partially funded by FAPERGS (17/2551-0001/205-4) and CNPq.

References

1. Adorjan, A., Matturro, G.: '24 hours of innovation'-a report on students' and teachers' perspectives as a way to foster entrepreneurship competences in engineering. In: World Engineering Education Conference, pp. 43–46. IEEE, Piscataway (2017)
2. Barbe, D.: A model of cross disciplinary education, technology transfer and teaching non-technical skills for engineers. In: Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments, Dublin, pp. 1–32. IEEE Computer Society, Piscataway (2010)
3. Bharadwaj, A.: An evaluation of teaching theoretical graduate engineering courses adapting different techniques. In: IEEE International Conference on MOOC, Innovation and Technology in Education (MITE), Patiala, pp. 84–88. IEEE Computer Society, Piscataway (2014)
4. Blank, S., Dorf, B.: The Startup Owner's Manual: The Step-by-step Guide for Building a Great Company. K&S Ranch, Incorporated, Pescadero (2012)
5. Boutell, M.R., Fisher, D.S.: Entrepreneurial minded learning in app development courses. In: Frontiers in Education Conference (FIE), pp. 1–8. IEEE, Piscataway (2017)
6. Breytenbach, J., de Villiers, C., Hearn, G.: Directing the South African ICT labour force towards growth sectors: a case for non-institutional scarce skills transition and reskilling courses. In: AIS Special Interest Group for Education: International Academy for Information Management – AIS SIG-ED IAIM 2013 International Conference on Informatics Education and Research Conference. Association for Information Systems, Milan (2013)
7. Buckley, M., Nordlinger, J., Subramanian, D.: Socially relevant computing. ACM SIGCSE Bull. **40**(1), 347–351 (2008)
8. Budgen, D., Turner, M., Brereton, P., Kitchenham, B.: Using mapping studies in software engineering. In: Proceedings of the 20th Annual Meeting of the Psychology of Programming Interest Group (PPIG 2008), pp. 195–204. Lancaster University, Lancaster (2008)
9. Buffardi, K., Robb, C., Rahn, D.: Learning agile with tech startup software engineering projects. In: Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (ITICSE 2017), pp. 28–33. ACM, New York (2017)
10. Buffardi, K., Robb, C., Rahn, D.: Tech startups: realistic software engineering projects with interdisciplinary collaboration. J. Comput. Sci. Coll. **32**(4), 93–98 (2017)
11. Case, S., Coleman, M., Deshpande, G.: The Innovative and Entrepreneurial University: Higher Education, Innovation and Entrepreneurship in Focus. US Department of Commerce, Economic Development Administration, Washington (2013)
12. Chanin, R., Sales, A., Pompermaier, L., Prikladnicki, R.: A systematic mapping study on software startups education. In: Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018, EASE'18, pp. 163–168. ACM, New York (2018)

13. Chanin, R., Sales, A., Santos, A., Pompermaier, L., Prikladnicki, R.: A collaborative approach to teaching software startups: findings from a study using challenge based learning. In: Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE '18, pp. 9–12. ACM, New York (2018)
14. Chenoweth, S.: Undergraduate software engineering students in startup businesses. In: Proceedings of the 21st Conference on Software Engineering Education and Training (CSEET'08), Charleston, pp. 118–125. IEEE Computer Society, Piscataway (2008)
15. Chesney, D.: Social context, singular focus. In: Proceedings of 2014 IEEE Frontiers in Education Conference (FIE), , Madrid, pp. 1–6. IEEE Computer Society, Piscataway (2014)
16. Coleman, G.: An empirical study of software process in practice. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS), Big Island, pp. 315c, 1–6. IEEE Computer Society, Piscataway (2005)
17. Coleman, G.: An empirical study of software process in practice. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005. HICSS'05, pp. 315c–315c. IEEE, Piscataway (2005)
18. Currie, E., Doboli, S., Kamberova, G.: Developing the next generation of entrepreneurs. In: Proceedings of 2011 IEEE Frontiers in Education Conference (FIE), Rapid City, pp. S2B 1–6. IEEE Computer Society, Piscataway (2011)
19. da Cruz, E.F.Z., Alvaro, A.: Introduction of entrepreneurship and innovation subjects in a computer science course in Brazil. In: 2013 IEEE Frontiers in Education Conference (FIE), pp. 1881–1887 (2013). https://doi.org/10.1109/FIE.2013.6685162
20. Daimi, K., Rayess, N.: The role of software entrepreneurship in computer science curriculum. In: Proceedings of the 2008 International Conference on Frontiers in Education: Computer Science & Computer Engineering (FECS 2008), Las Vegas, pp. 332–338. IEEE Computer Society, Piscataway (2008)
21. de Lange, P., Nicolaescu, P., Klamma, R., Koren, I.: DevOpsUse for rapid training of agile practices within undergraduate and startup communities. In: European Conference on Technology Enhanced Learning, pp. 570–574. Springer, Heidelberg (2016)
22. Devadiga, N.M.: Software engineering education: converging with the startup industry. In: 2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T), pp. 192–196. IEEE, Piscataway (2017)
23. Engelsma, J.: Best practices for industry-sponsored CS capstone courses. J. Comput. Sci. Coll. **30**(1), 18–28 (2014)
24. Fagerholm, F., Hellas, A., Luukkainen, M., Kyllönen, K., Yaman, S., Mäenpää, H.: Patterns for designing and implementing an environment for software start-up education. In: Proceedings of the 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2017), pp. 133–140. IEEE, Piscataway (2017)
25. Ford, R., Goodrich, J., Weissbach, R.: A multidisciplinary business and engineering course in product development and entrepreneurship. In: Proceedings of the 34th Annual Frontiers in Education (FIE 2004), Savannah, pp. T2E/5–T2E10. IEEE Computer Society, Piscataway (2004)
26. Génova, G., González, M.: Educational encounters of the third kind. Sci. Eng. Ethics **1**, 1–10 (2016)
27. Giardino, C., Unterkalmsteiner, M., Paternoster, N., Gorschek, T., Abrahamsson, P.: What do we know about software development in startups? IEEE Software **31**(5), 28–32 (2014)
28. Giardino, C., Wang, X., Abrahamsson, P.: Why early-stage software startups fail: a behavioral framework, pp. 27–41. Springer International Publishing, New York (2014)
29. Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: the greenfield startup model. IEEE Trans. Softw. Eng. **42**(6), 585–604 (2016)
30. Gross, W.: An approach to teaching entrepreneurship to engineers. In: Proceedings of the 2000 IEEE Engineering Management Society (EMS 2000), Albuquerque, pp. 648–652 (2000)

31. Harms, R.: Self-regulated learning, team learning and project performance in entrepreneurship education: learning in a lean startup environment. Technol. Forecast. Soc. Chang. **100**, 21–28 (2015). https://doi.org/10.1016/j.techfore.2015.02.007
32. Heintz, F., Klein, K.I.: The design of Sweden's first 5-year computer science and software engineering program. In: Proceedings of the 45th ACM Technical Symposium on Computer Science Education, Atlanta, pp. 199–204 (2014)
33. Izurieta, C., Trenk, M., O'Bleness, M., Gunderson-Izurieta, S.: The effectiveness of software development instruction through the software factory method for high school students. In: 123rd Annual Conference in Engineering and Education (ASEE'16), New Orleans, pp. 26–29 (2016)
34. Järvi, A., Taajamaa, V., Hyrynsalmi, S.: Lean Software Startup – An Experience Report from an Entrepreneurial Software Business Course, Braga, pp. 230–244. Springer International Publishing, New York (2015)
35. Joseph, A.: Interdisciplinarity, financial software product development, and entrepreneurship in an urban university. Am. Soc. Eng. Edu. **11**(1), 812.1–812.13 (2006)
36. Kajko-Mattsson, M., Nikitina, N.: From knowing nothing to knowing a little: experiences gained from process improvement in a start-up company. In: International Conference on Computer Science and Software Engineering (CSSE 2008), Wuhan, pp. 617–621 (2008)
37. Kaltenecker, N., Hoerndlein, C., Hess, T.: The drivers of entrepreneurial intentions – an empirical study among information systems and computer science students. In: Proceedings of the 19th Americas Conference on Information Systems (AMCIS 2013), Chicago, pp. 1–8 (2013)
38. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, Keele and Durham (2007)
39. Kitchenham, B., Budgen, D., Brereton, O.: Using mapping studies as the basis for further research – a participant-observer case study. Inf. Softw. Technol. **53**(6), 638–651 (2011)
40. Ko, A.J.: A three-year participant observation of software startup software evolution. In: Proceedings of the 39th International Conference on Software Engineering: Software Engineering in Practice Track, pp. 3–12. IEEE Press, Piscataway (2017)
41. Kontio, J., Ahokas, M., Poyry, P., Warsta, J., Makela, M., Tyrvainen, P.: Software business education for software engineers: towards an integrated curriculum. In: 19th Conference on Software Engineering Education and Training Workshops (CSEETW'06), pp. 4–7 (2006). https://doi.org/10.1109/CSEETW.2006.15
42. McMahon, E.: From product development to innovation. In: Proceedings of the International Annual Conference of the American Society for Engineering Management (ASEM 2014), Virginia Beach, pp. 118–127 (2014)
43. Nguyen-Duc, A., Seppänen, P., Abrahamsson, P.: Hunter-gatherer cycle: a conceptual model of the evolution of software startups. In: Proceedings of the 2015 International Conference on Software and System Process (ICSSP 2015), Tallinn, pp. 199–203 (2015)
44. Nguyen-Duc, A., Shah, S., Ambrahamsson, P.: Towards an early stage software startups evolution model. In: Proceedings of the 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2016), Limassol, pp. 120–127 (2016)
45. Nurkkala, T., Brandle, S.: Software studio: teaching professional software engineering. In: Proceedings of the 42nd ACM Technical Symposium on Computer Science Education, Dallas, pp. 153–158 (2011)
46. Osterwalder, A., Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. John Wiley & Sons, Hoboken (2010)
47. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: a systematic mapping study. Inf. Softw. Technol. **56**(10), 1200–1218 (2014)
48. Pauca, V., Guy, R.: Mobile apps for the greater good: a socially relevant approach to software engineering. In: Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE '12), Raleigh, pp. 535–540 (2012)

49. Pauli, J., Lawrence, T., Brown, B.: Development of a new software product from a classroom project. In: Proceedings of the 5th International Conference on Information Technology: New Generations (ITNG 2008), Las Vegas, pp. 97–100 (2008)
50. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), Bari, pp. 68–77 (2008)
51. Porter, J., Morgan, J., Lester, R., Steele, A., Vanegas, J., Hill, R.: A course in innovative product design: a collaboration between architecture, business, and engineering. In: Proceedings of 2015 IEEE Frontiers in Education Conference (FIE), El Paso, pp. 1–5. IEEE Computer Society, Piscataway (2015)
52. Porter, J., Morgan, J., Lester, R., Steele, A., Vanegas, J., Hill, R.: A course in innovative product design: a collaboration between architecture, business, and engineering. In: 2015 IEEE Frontiers in Education Conference (FIE), pp. 1–5. IEEE, Piscataway (2015)
53. Quezada-Sarmiento, P.A., Enciso, L., Mayorga-Diaz, M.P., Mengual-Ándres, S., Hernandez, W., Vivanco-Ochoa, J.V., Carrión, P.V.: Promoting innovation and entrepreneurship skills in professionals in software engineering training: an approach to the academy and bodies of knowledge context. In: 2018 IEEE Global Engineering Education Conference (EDUCON), pp. 796–799. IEEE, Piscataway (2018)
54. Ribeiro, C., Aleixo, F., Freire, M.: Driving academic spin-off by software development process: a case study in federal Institute of Rio Grande do Norte-Brazil. In: Proceedings of the 17th International Conference on Product-Focused Software Process Improvement (PROFES 2016), Trondheim, pp. 636–639 (2016)
55. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business, New York (2011)
56. Rio, C.R.D., Morgado-Estevez, A., Dominguez-Jimenez, J.: Entrepreneurship and lean manufacturing for software engineering. In: SEFI Annual Conference 2014, Birmingham (2014)
57. Rubin, J., Chisnell, D.: Handbook of Usability Testing: How to Plan, Design and Conduct Effective Tests. John Wiley & Sons, Hoboken (2008)
58. Salas, R.P.: Teaching entrepreneurship in computer science: lessons learned. In: Frontiers in Education Conference (FIE), pp. 1–7. IEEE, Piscataway (2017)
59. Salleh, N., Mendes, E., Grundy, J.: Empirical studies of pair programming for CS/SE teaching in higher education: a systematic literature review. IEEE Trans. Softw. Eng. **37**(4), 509–525 (2011)
60. Sarraipa, J., Ferreira, F., Marcelino-Jesus, E., Artifice, A., Lima, C., Kaddar, M.: Technological Innovations tackling Students dropout. In: Proceedings of the 7th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion, Vila Real, pp. 112–118. ACM, New York (2016)
61. Schilling, J., Klamma, R.: The difficult bridge between university and industry: a case study in computer science teaching. Assess. Eval. High. Educ. **35**(4), 367–380 (2010)
62. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall PTR, Upper Saddle River (2001)
63. Shaw, M.: Writing good software engineering research papers: minitutorial. In: Proceedings of the 25th International Conference on Software Engineering (ICSE '03), Portland, pp. 726–736. IEEE Computer Society, Piscataway (2003)
64. Sun, D., Xue, J., Tan, X., Liu, P., Sun, Z., Yao, J.: Model analysis of talents' abilities and qualities for information-based entrepreneurship. In: Proceedings of the 1st International Conference on Information Science and Engineering (ICISE 2009), Nanjing, pp. 2968–2971 (2009)
65. Vitolo, T., Hersch, K., Brinkman, B.: Making the connection: successful cross campus collaboration among students. In: Proceedings of 2016 IEEE Frontiers in Education Conference (FIE), Erie, pp. 1–7. IEEE Computer Society, Piscataway (2016)
66. Wieringa, R., Maiden, N., Mead, N., Rolland, C.: Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requir. Eng. **11**(1), 102–107 (2005)

67. Zaina, L., Alvaro, A.: A design methodology for user-centered innovation in the software development area. J. Syst. Softw. **110**(C), 155–177 (2015)
68. Zhang, S.: A technology-business-environment model for effective internet entrepreneurship education. In: Proceedings of the 12th International Conference on Information Technology-New Generations (ITNG 2015), Las Vegas, pp. 632–637 (2015)

Teaching "Through" Entrepreneurship: An Experience Report

Xiaofeng Wang, Dron Khanna, and Marco Mondini

Abstract Entrepreneurship education varies from theoretical courses that take a "teacher-centered" perspective to more hands-on, "learning by doing" ones that put students to the center of the attention and engage them in acquiring entrepreneurship competencies through experience. In this chapter, we present our experience of teaching the Lean Startup methodology to university students from diversified academic background. Rather than describing our teaching experience in the past years as a whole, we focus on a particular interesting part of our experience, which is, while we teach students Lean Startup and guide them to work on real startup projects during the course, we were also developing our own startup idea related to the course—Startuppuccino, a startup education platform to support active entrepreneurial learning. Through this unique experience, we reflect on how to better organize the course and enhance the active learning of the students using a supportive platform.

Keywords Entrepreneurship education · Lean startup · Learning by doing · Unique value proposition

1 Introduction to Entrepreneurship Education

Entrepreneurship is important for the process of value creation, job creation, and general economic development [1]. Higher education institutions are expected to teach entrepreneurship and produce graduates with a broad set of enterprising competencies and skills, and ambitions to become entrepreneurs [2]. Teaching entrepreneurship at the university level can also stimulate the research on entrepreneurship [3].

However, entrepreneurship education is a deep, diversified, broad, and interdisciplinary field about which we just started to apprehend as software engineering

X. Wang (✉) · D. Khanna · M. Mondini
Free University of Bozen-Bolzano, Faculty of Computer Science, Bolzano, Italy
e-mail: xiaofeng.wang@unibz.it; dron.khanna@unibz.it

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_14

researchers. Entrepreneurship is taught from various theoretical viewpoints. Moreover, educating objectives during the course and teaching approaches differ notably. This is due to the fact that entrepreneurship is taught within various faculties by academic with different backgrounds and skills, to students with all kinds of educational backgrounds [1].

With the increasing appreciation for the significance of entrepreneurship education and interest in teaching entrepreneurship, various questions also emerge: Could entrepreneurship be taught? Who are the students? Why are the students taking the course? What is the expected learning outcome? and How could the entrepreneurship courses be taught effectively? [4, 5] A broad variety of practices and studies provide various responses. Based on a systematic literature review, Sirelkhatim and Gangi [6] summarize three generic themes of entrepreneurship education provision:

1. Teach "about" entrepreneurship: courses that are theory oriented and aim to increase awareness about entrepreneurship and instill various entrepreneurship knowledge to students
2. Teach "for" entrepreneurship: practical courses that aim to encourage students and enhance their intentions to be entrepreneurs in future. They generally take skill-based approaches where they seek to train students about the mechanisms of running a business
3. Teach "through" entrepreneurship: practical courses which aim to graduate entrepreneurs, support new venture creation, and develop entrepreneurial competencies. Even though the curricula content for this theme may be similar to teaching "for" entrepreneurship, it suggests learning "with" and "through" real-life entrepreneurship, to enable students to experience "being" entrepreneurs rather than "pretending" to be ones and to have a real taste of market forces.

In the past 7 years, we have been teaching the Lean Startup methodology to university students from diversified academic background. The Lean Startup approach was inspired by Lean concepts of focusing on the efforts that create value to customers and eliminating waste during entrepreneurial processes [7]. However, since the customers are often unknown, what customers could perceive as value is also unknown. Therefore, entrepreneurs should "get out of the building" to involve the customers since day one. Lean Startup advocates to build the product iteratively and deliver to the market for earlier feedback [7]. Therefore, Lean Startup is essentially a hypothesis-driven approach [8] which bases entrepreneurial decisions on evidence and validated learning. To capture customer value, an entrepreneur should start a feedback loop that turns an idea into a product and then learn whether to pivot or persevere. This can be done by developing a minimum viable product (MVP) using agile methods to collect customer feedback on the product [7]. The feedback becomes the input to improve the product and validate the hypotheses. As the result, the startup might pursue a new direction of the business or continue and scale it.

The experience presented in this chapter is an example of the application of the teaching "through" entrepreneurship method. What makes this experience unique is that the method was applied in double senses: the participating students worked on real startup projects during the course, learning the Lean Startup knowledge by doing, and the teaching team, apart from teaching, was also developing our own startup idea related to the course—Startuppuccino, a startup education platform, using the same Lean Startup methodology that we are teaching. This chapter is a recount of this interesting journey, and the reflection on how to better organize the course and enhance the active learning of the students using a supportive platform.

2 The Lean Startup Course

The Lean Startup course offered at our university is a semester long course open to the students from all faculties of the university (computer science, design and art, economics, science and technology as well as education) at all levels (undergraduate, master, and PhD). The number of students ranges from 30 to 60. The teaching team is composed of 2–3 faculty staff. In addition, a group of mentors are also involved in the course, who are experienced entrepreneurs or work in the related domains, and volunteer to help the students to develop their ideas. Based on the number of enrolled students and the manner in which mentors are involved in the course, the number of mentors could vary from 6 to 14. The course is project based, problem driven, and mainly learning by doing. The students propose original business ideas, form project teams based on chosen ideas, and develop them into a business as far as they can during a semester. There is a minimal amount of upfront lecturing and on-demand seminars on the topics that students frequently inquire or request. The topics covered in the theoretical part include the definition of a startup, ideation, problem/need identification and validation, build-measure-learn loop, MVP, lean analytics, lean canvas, building entrepreneurial teams, getting funded, etc.

The structure of the course has stabilized through refinement over the years. Figure 1 is a representation of the timeline of the student projects.

Each student startup project will go through roughly four phases: (1) Ideation, (2) Problem Validation, (3) Problem-Solution Fit, and (4) Pitch Preparation. Naturally the process is more iterative than linear. A student team might go through these phases multiple times if they pivot their original ideas. Each phase will end with a corresponding deliverable, which allows the teaching team to analyze the progress of the projects and provide early and continuous feedback. The feedback sessions are organized as team retrospectives, which are the meetings that the teaching staff arrange with each individual student teams.

The most important milestone of the project is the pitch competition event toward the end of the course. It is not only an event at which the student teams could present their startup projects to the public audience. It is also considered the final exam of the course. The final team grades are partially based on the performance of the teams at the event. The students do not receive individual scores.

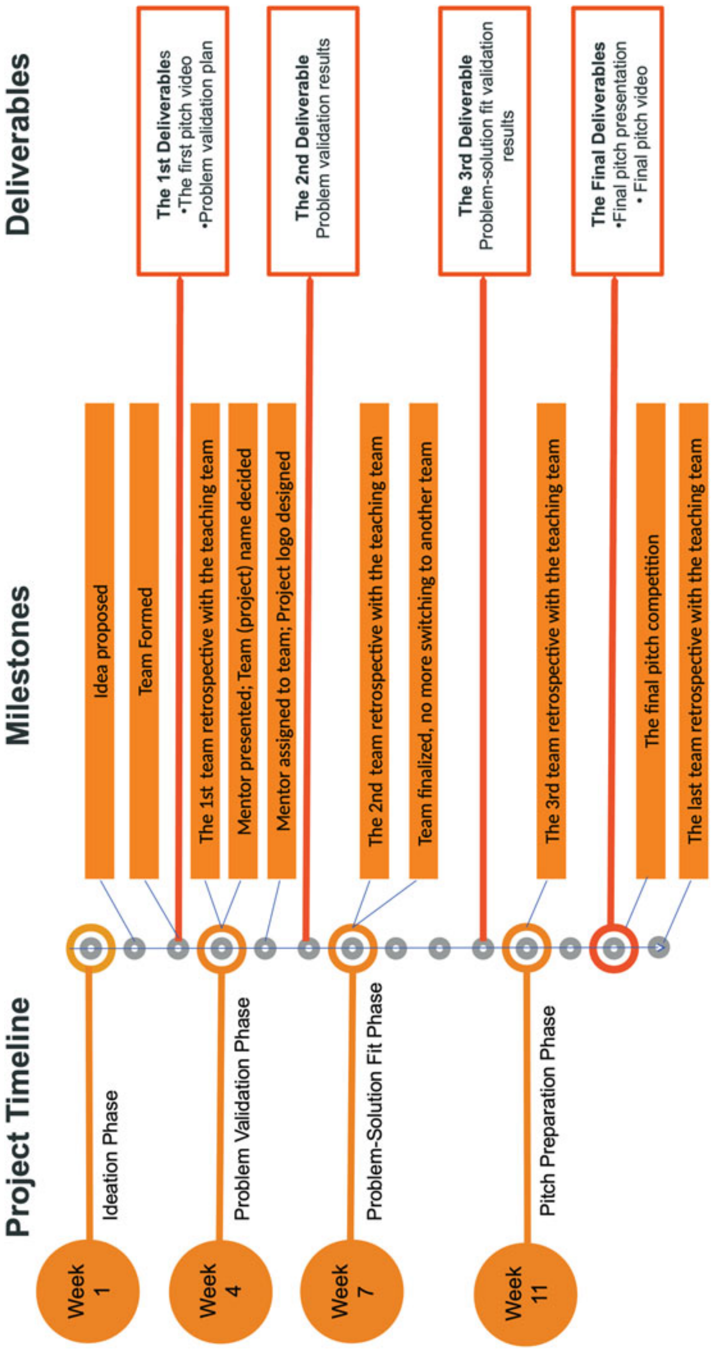

Fig. 1 The project timeline at the Lean Startup course

The course ends with a final retrospective session with each project team. Typically during these final retrospective sessions, we ask the students to recall the journey they went through. The team would talk about how they feel their project went. It is an open session with the aid of the so-called "mood chart" to facilitate the conversations between the teachers and the students (Fig. 2 is an illustration of a mood chart).

The final retrospectives serve multiple purposes: (a) for the teaching team to obtain better visibility to the work and effort of each student team; (b) for the teaching team to collect feedback from the students regarding the course; (c) for the students to have an opportunity to reflect and enhance the learning; and (d) for the students to have a sense of closure of the course and their project, as the course journey can be a "roller coaster" emotionally for the teams if they take the project sufficiently serious.

3 The Startuppuccino Journey

Over several years of teaching Lean Startup to the university students and interacting with local startups and entrepreneurs, the teaching team has observed that there was a lack of support to early-stage startups, especially student startup teams who badly need guidance and support on how to turn their ideas into concrete business. The initial idea of the teachers was to recommend good software tools to initiate and support startups that are missing key skills in their teams (e.g., design and web development). After the discovery of several first-moving competitors and a revealing experience by participating in a local Startup Weekend event, the teaching team discovered that early-stage startups need to acquire basic knowledge about how to build a startup before they could realize what tools they need and how they can benefit from them. Based on this realization, the team decided to pivot to providing direct guidance to entrepreneurs by connecting them to a pool of mentors. However, the team proceeded very slowly toward this identified direction. No real development activity commenced. One key reason was that none of the team members had experience of community building, and the team was not in a position to do it since it required rich connections with local startup ecosystem players. The small pool of mentors we built up through the Lean Startup course was far from sufficient.

After several brainstorming sessions and realignment of the visions of different members, the team decided to focus on one thing that the team knew well and knew what potential pain points they could tackle: bring active learning to student entrepreneurs through better supporting the work of educators. This was the moment of birth of Startuppuccino, the startup idea that the teaching team pursued using the very Lean Startup approach we taught to the student teams, and the validation of the idea was closely tied back to the Lean Startup course.

Fig. 2 The Mood chart used to conduct final retrospective with the student teams

3.1 Just-in-Time Development of the Minimum Viable Platform

In the next several months following the discovery of the new direction, the team had intensive brainstorming sessions to share the new vision and to figure out how to start again in a lean manner. Customer journey mapping, a tool from Design Thinking, has been applied to get the understanding of the scope and initial set of requirements. However, the project was somehow stagnant, and no significant and exciting progress has been made. The team needed a push to make significant progress, and soon realized that there was one coming soon—the teaching team is going to start teaching the new edition of the Lean Startup course. It would be a real and optimal opportunity to start the experiment of our idea. At that moment, the team believed that a "concierge MVP" (Ries [7]) is a correct way to go forward. The teaching team themselves are the "concierges," the chosen customers, and the team would gain a better understanding of the problem that Startuppuccino intends to tackle as well as the solution it could offer.

To implement the concierge MVP, the team needed to build a minimum viable platform to be able to gather feedback from the students. The assumption the team made was that, even though educators were the intended customers, students have to see value in the platform and be willing to use it before educators can benefit from it. However, the team had to speed up the initial development of the platform in order to start the build-measure-learn loops. There were two choices: (1) develop very intensively during a short period and have a running platform ready to use in the beginning of the course; or (2) take a more stepwise approach and develop the application along the course. Partially due to the limited development resources and time the team had, but more importantly due to the conviction the team had on developing in an agile and lean manner to avoid waste in developing something that is not going to be useful for the users, the team decided to go with the second choice.

Since the course had weekly sessions, the team adopted weekly iterations too. In order to decide what to be developed in each iteration and to prioritize the tasks, the argument of developing MVP has been used extensively. The team members constantly reminded each other that we were developing a minimum viable platform that should allow us to understand if students would use such a platform with the support of educators. Any addition of the new functionality was evaluated using the criteria of minimum and viable.

One decision made purposefully by the team that reflects the team's understanding of MVP was not to develop the graphical user interface for educators, despite the shared understanding that they would be the paying customers of Startuppuccino. The teachers from the team modified database tables directly when certain changes were needed from the perspective of educators.

This just-in-time but intensive manner of development put much pressure on the developers to deliver the working functionality needed for each week. However, the benefit was visible too. We got immediate feedback from the students and other intended users (e.g., mentors that supported student projects) on the developed functionality, and if and what to develop more in the following week.

3.2 Continuous Experimentation Guided by the Unique Value Proposition

The intensive development of the minimum viable product gave the team a sense of achievement and satisfaction. However, it also obscured the team's vision of a more crucial aspect of the startup: What are the value propositions [9] that the platform provides to the users and customers? How unique are they? [10] and How are we different from other existing platforms to support entrepreneurship courses?

In the middle of the course, after the prototype development came into a stable stage, the team had the opportunity to support another course in another university, even though it was not an entrepreneurship course per se. The team failed to support this course due to the limited capacity at that moment. However, a big debate happened within the team as to whether we should support non-entrepreneurship courses, which turned out to be crucial for the team to question what were the unique value propositions (UVPs) of Startuppuccino. Several brainstorming sessions were dedicated to answer this question. Eventually the team agreed that providing the visibility of the learning of the student teams to the educators is the unique value proposition that we should focus on. This UVP became the compass that guided the subsequent development of Startuppuccino.

With this new understanding, the team realized that we needed to design new MVPs to find the answers to two key questions: (1) How to capture the learning from students? and (2) How to make it visible to educators?

Right from the beginning the team knew that, for entrepreneurship courses which generally have a prevalence of practical activities, a "learning by doing" style, conventional intermediate and final exams are not the appropriate devices to capture the true learning that students could achieve. The teachers of the course borrowed the practice of retrospectives from agile methods. As shown in Fig. 1, each student team has been required to attend several private retrospective sessions with the teaching staff during and after the course. Originally the main goal of the retrospectives was for the teachers to get a clear idea on the status of all projects. Soon the Startuppuccino team realized that it could be a good occasion for the teachers to understand if the students have learnt properly. However, several drawbacks of this approach also emerged. The students tended to forget about the experience if a retrospective session happens too late. And it was time consuming and resource intensive for the teachers to run retrospective sessions with all student teams (10–15 teams). Based on this observation, the Startuppuccino team realized that a feature that can support continuous retrospectives of student teams by themselves would achieve the goal of capturing learning more accurately and save the time and energy of educators.

In order to act quickly, the team applied the concept of "piecemeal MVP" from the Lean Startup approach. Rather than developing a native form in the platform, which takes more time, the team quickly designed a Google form, which allowed us to test what should be collected from the students and how to collect them. This form has been our MVP to validate the risky assumptions that "educators can evaluate the learning of students with the information collected by the form," and "students are able to interpret and provide valuable information through the questions asked in the form." After being instructed on how to use the form, the students had not been forced to fill the form. In the end, we collected only few answers, and even less of them that give a good visibility of the students' learning to educators. The experiment result with the Google form MVP was not positive, but it still offered some validated learning. The team realized that, in order to gather the learning from students regularly, a form only may not be sufficient.

Meantime, to understand how to visualize the learning has proven to be even more difficult. The team went back to the retrospective charts drawn by the students from the previous courses. However, since the teachers have been giving students instructions on how to draw the retrospective charts (e.g., draw a timeline of what happened), the team was not sure if this was really the correct way to visualize the learning. In order to experiment on the visualization part, when the current edition of the course ended, the teaching team did final retrospectives with each student team and left open to the students to illustrate their learning on flipcharts. After all charts were drawn, the team analyzed them collectively to identify common elements and to evaluate their effectiveness of representing learning. One main validation from this experiment was that the majority of the student teams did actually follow a timeline flow even when they were left open to draw their learning in any format. We also asked the members of the team that were less exposed to the course to interpret the charts, to see which ones were easier for them to understand the project and learning points. The charts that followed a timeline were easier to interpret.

At this point, the team started to design the learning chart and its components that would be eventually implemented in the Startuppuccino platform. In order to implement this part in a lean manner, we needed new opportunities to experiment the solution. Since we were running out of time and students, we decided to consider our own startup experience with Startuppuccino in order to test our design of "collection and visualization of the learning." Mockup MVP is the device we used to experiment the solution. We used flipchart with post-it notes to simulate the interface of online learning chart. As a result, we created our own learning chart (see Fig. 3. Post-it notes with green text: What happened; Post-it notes with red text: What we have learnt).

At the end of this experiment all the team members were satisfied with the experience and the general feeling was a happy surprise from the work and achievement we obtained. The feeling of being surprised was due to the memory of many crucial events forgotten by any single team member. The experiment on ourselves showed us the value of having a place, in this case the A1-sized flipchart, to collect all the information, and the need for collaboration of all the team members

Fig. 3 The learning chart of Startuppuccino produced during a retrospective session

to build the learning chart. It also led us to the realization that the Google form MVP could be useful only in the context of team retrospectives.

However, we were aware of the limitation of this experiment using our own experience, and the possibility of biases. Moreover, it was a one-off rather than continuous retrospective intended by our solution. This was the reason why we looked for more opportunities to run the full experiment. A new opportunity came along when two team members were invited to run a 1-week intensive Lean Startup course in a large Brazilian university. At the end of each day during this intensive teaching week, the students were asked to conduct the retrospective and provide the learning using the Mockup MVP we provided. We observed how the students used the Mockup MVP to create their learning chart, if they followed our instruction, and collected the feedback from them. The result experience was positive, and allowed us to learn about more detailed aspects of our solution. One main validation came from two students who were professors in another institute. They found that the learning chart was a very good way to understand students' learning, and told us that they would go back to try with their students.

Based on the learning from this set of experiments guided by the UVP, the team came to the realization that the most valuable feature of the Startuppuccino platform would be the online learning chart, which would be developed in the future iterations.

4 Lessons Learnt

Most important lessons learnt:

- The entrepreneurial mentality is a more important learning outcome than the realisation of a startup idea in entrepreneurship education.
- Teaching by doing enables the educators to better implement learning by doing for the students.
- A diversified team means more diversified ways of thinking and problem solving than diversified knowledge and skills.

In this section we reflect on the experience reported in the previous sections and highlight the key lessons learnt.

4.1 The True Objective of Teaching "Through" Entrepreneurship

When we started the course, we believed that the main objective of the course was to help the students to realize their business ideas by instilling the entrepreneurial knowledge they need and provide the support as much as possible, such as mentoring, missing skills, etc. This is in line with the teaching "through" entrepreneurship method we adopted.

Correspondingly, the way we evaluated if we were successful in teaching Lean Startup was how many projects became real startups. However, the number was disappointing, and we were confused of the role we played in comparison to startup incubators, and we were not able to distinguish our course from the local incubation programs and initiatives, such as Startup Weekend events.

With accumulated experience and interaction with students, and our own startup experience with Startuppuccino, we came to the realization that the true objective that is more in line with the nature of university course and participating students should be nurturing the entrepreneurial spirits in the students, which would be beneficial in many ways in the future no matter what they do and if they are going to start their companies or work for others. This objective shift is not trivial in the sense it would affect how the course should be designed, students be guided, and resources be used. Neither is it in contradiction with the teaching "through" entrepreneurship method, as the right mentality cannot be formed without real action as well as the reflection upon the experience.

4.2 Learning by Doing and Teaching by Doing

Learning by doing is an attractive aspect of the Lean Startup course and the students generally responded positively to this style, stating during the retrospectives that this course was completely different as compared to other courses held at the university, and they have learnt a lot and enjoyed the course. However, the learning outcome can be quite different for students with different attitudes toward the course. As we always communicated with the students, the learning outcome of each individual highly depended on how motivated, committed, and proactive he/she was. Having said so, we as educators have the responsibility to facilitate the students to obtain good learning experience.

In this sense, Startuppuccino is our attempt as teachers to ensure and improve the learning that the students can obtain from learning-by-doing courses. Through the engagement with the course and classmates using the platform, the students get properly motivated, timely supported, and encouraged to continuously reflect on their experience to draw valid learning points.

In addition, we have obtained our own learning by doing through developing Startuppuccino. Four main points, valuable to us, can be useful to the student entrepreneurs as well as any early-stage startups.

(1) Focus on what the team knows the best, and acquire the new knowledge along the way.

Originally we wanted to build a community supported by an online platform, which could connect early-stage startup companies and entrepreneurs with the resources and guidance they need. However, community building demands the expertise that the team did not have and were not familiar with. As a consequence, the team moved slowly and lacked momentum to make real progress. After pivoting to the idea of building an active learning environment to support educators to better serve student entrepreneurs, the team found their home ground. Two key members of the team are educators themselves. They run entrepreneurial courses and know what students like and do not like about the courses. The pain points that the team targets at are the pain points felt by them directly. Starting with what the team knows best, we could quickly identify where and how to proceed in a more confident manner.

(2) Need for a strong business driver.

The experience of developing the minimum viable platform described in Sect. 3 made us realize the need and effectiveness of having a strong business driver. Even though we decided and started with what we knew the best, the speed of the development still suffered until we had identified one business case, which was to support the upcoming entrepreneurship course. The course provided a sense of real business and urgency, which put the team into real action and motivated us to produce tangible results. The two educators acted as super early adopter of the idea and on-site customer of the solution development, and provided continuous drive for the team to push the idea forward.

(3) *Obtain unique value proposition along the way.*

In retrospect, the main purpose of the concierge MVP we created was to start the learning loop. It was developed without a good understanding of what the unique value proposition of our startup was. However, it does not mean the effort of developing the minimum viable platform was a waste. We would not have been able to achieve the understanding of our UVP without doing it. In this sense, unique value propositions are not obtained upfront, but along the way of continuous experimentation. A clearly identified UVP provides a good target and guidance for experiments with MVPs. After we understood what our UVP is, we had been able to utilize all the possible opportunities we could grab to conduct a set of experiments, with a set of MVPs.

(4) *Grab any opportunity or create it to experiment with MVPs.*

We utilized different MVPs to conduct the experiments that allow us to obtain validated learning on our business idea before investing significantly to develop the full solution. We started with the concierge MVP with the support of a minimum viable platform that we developed in a just-in-time manner with the help of several agile practices. We learnt the value in piecemeal delivery that allowed us to obtain constant feedback and adjust our course accordingly. In addition, as a small startup with limited resources, we need to seize every possible opportunity, and even proactively create new opportunities, in order to conduct the experiments with MVPs that allow us to obtain validated learning. We need to experiment, reflect, and learn in a continuous manner. Validated learning is the true measurement of the progress of an early-stage startup, and one would not be able to obtain validated learning without experimenting and reflecting constantly.

4.3 Team Formation

We have advocated that an ideal entrepreneurial team should be composed of members of different knowledge, skills, and background, based on the understanding that diversity is a key attribute for innovative ideas [11]. Especially we encouraged the students to form the team in such a way that the competence such as design, software development, and marketing is in the team. Especially the important role of design in the Lean Startup course projects has been demonstrated repeatedly in various tasks.

However, the diverse skills brought by different team members should not define the types of contribution they could bring to the development of a startup project. Too often we saw that the project teams only used the skills of their team members, but they were not benefiting from different perspectives and ways of thinking enabled by different disciplines and educational backgrounds. That was why we kept reminding the student teams that computer science students should contribute their technical knowledge to their projects rather than just code, and the designers were part of a team not just because they were good at designing logos. Our own

Startuppuccino team has a good mix of various types of knowledge and skills, and we have tried our best to use “the brains” rather than just “the hands” of everyone.

4.4 Mentoring

Mentors and mentoring are one cornerstone of our Lean Startup course. Students can get direct learning and valuable feedback and guidance through interacting with different mentors. We have experimented (and failed) various ways of engaging the mentors and ensuring the quality of mentoring. Following is the solution we found effective:

– After project teams are formed, a two-way matching between teams and mentors is conducted to identify which mentor helps which team; mentors can vote up to three projects that they prefer to help, and teams three mentors based on the expertise that they need;
– Each mentor takes the ownership of the project team matched to him/her, and mentor that project during the whole semester; and
– To allow the whole class benefit from the knowledge of different mentors, every week we appoint at least two “Mentors of the Week” and for that week the appointed mentors have the responsibility to mentor all the projects and answer all the questions.

This arrangement got positive reaction from the students. However, one issue we are still struggling with is how to handle the situations where different mentors have different levels of commitment and engagement with the course, which affects directly the learning experience of students.

4.5 Assessment of Student Teams

To grade the Lean Startup projects and students has always been a difficult task for us the teaching team, both technically and emotionally, even though we made clear to the students right from the start of the course that each project team would get one unified score and no different scores would be given to the members of the same team. Our rationale was that this way of grading would encourage teamwork which was crucial for project-based courses. It also simulated the real-world situations where the team would succeed or fail together. However, despite the fact that the students accepted this way of grading without complaint, we were aware that the students were not treated fairly from the point of view of the grades they received, which may affect their academic performance if undeserved low scores were given.

No complete visibility to the effort the teams put into the projects was the main reason that weakened our confidence with the scores we gave to the project teams. We have used the regular retrospective meetings with the student teams to gain the visibility to their project progress and efforts. We have also used Startuppuccino to further improve and document the project work of the students. However, to achieve complete visibility desired by the teaching team is time and effort consuming, given the diverse nature of the ideas that the student teams are working on and their different teamwork styles and learning path. This is a challenge that the teaching team need to confront with better theoretical knowledge and practical tool support.

5 Conclusion

Lean Startup has become a prevailing approach applied by many startups in the past years. At the core of the approach is using build-measure-learn loops that allow a startup to quickly validate its business idea and to test different aspects of their business models, in order to acquire validated learning. As Ries argues [7], validated learning is the true measurement of the progress a startup makes, especially at its early stage.

In this chapter, we described a Lean Startup course taught at our university, together with our own experience of implementing a startup idea implemented as Startuppuccino. Especially our Startuppuccino experience reveals that MVPs play crucial and multiple roles in a startup life. To develop right MVPs to get validated learning, a startup needs to understand its unique value proposition. Neither could be achieved without continuous experimenting, reflecting, and learning from the team.

This learning could be incorporated in the courses that employ the teaching "through" entrepreneurship method. We hope that our lessons learnt can be valid input for those who would set up Lean Startup teaching in their institutions, and for those who follow the Lean Startup approach in developing their business ideas.

References

1. Blenker, P., Elmholdt, S.T., Frederiksen, S.H., Korsgaard, S., Wagner, K.: Methods in entrepreneurship education research: a review and integrative framework. Educ. Train. **56**(8/9), 697–715 (2014)
2. Gibb, A., Hannon, P.: Towards the entrepreneurial university. Inter. J. Entrepr. Educ. **4**(1), 73–110 (2006)
3. Fayolle, A., Kickul, J.: New and emerging perspectives for future research in entrepreneurship education. In: Fayolle, A. (ed.) Handbook of Research in Entrepreneurship Education, vol. 2, pp. 1–10. Edward Elgar, Northampton (2007)
4. Brand, M., Wakkee, I., Van Der Veen, M.: Teaching entrepreneurship to non-business students: Insights from two Dutch universities. In: Handbook of Research in Entrepreneurship Education, vol. 2, pp. 52–83. Edward Elgar, Northampton (2007)
5. Blenker, P., Korsgaard, S., Neergaard, H., Thrane, C.: The questions we care about: paradigms and progression in entrepreneurship education. Ind. High. Educ. **25**(6), 417–427 (2011)
6. Sirelkhatim, F., Gangi, Y.: Entrepreneurship education: a systematic literature review of curricula contents and teaching methods. Cogent Bus. Manage. **2**(1), 1052034 (2015)

7. Ries, E.: The lean startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Books, New York (2011)
8. Eisenmann, T., Ries, E., Dillard, S.: Hypothesis-driven entrepreneurship: the lean startup. In: Harvard Business School Entrepreneurial Management Case No. 812–095 (2012)
9. Osterwalder, A., Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. Wiley, Hoboken (2010)
10. Maurya, A.: Running Lean: Iterate from Plan A to a Plan that Works. O'Reilly Media, Sebastopol (2012)
11. Post, C., De Lia, E., DiTomaso, N., Tirpak, T.M., Borwankar, R.: Capitalizing on thought diversity for innovation. Res. Technol. Manage. **52**(6), 14–25 (2009)

Lean Internal Startups: Challenges and Lessons Learned

Henry Edison

Abstract To compete in this age of disruption, large and established organizations cannot rely on the traditional way of advancement, which focuses on cost efficiency, lead time reduction, or quality improvement. They are now looking for new ways to innovate like startups. With greater resource in-house, they hope that they can bring innovative products to market with new customer value as startups do. Along with it, the awareness and adoption of the Lean startup approach have grown rapidly amongst large and established organizations in recent years. While Lean startup method is originated from the software startup community, the core ideas behind Lean startup can offer benefits for large companies as well. If the obstacles can be minimized, the opportunities can be very beneficial to leverage software product innovation. This chapter illustrates some of the typical challenges that were met during real-world new Lean internal startups, and how they were solved.

Keywords Lean startup · Internal startup · Lean internal startup · Large organization

1 Introduction

Today, software startups have become one of the key drivers of economy and innovation. Research shows that high-growth startups account for 50% of new jobs created [14]. They differentiate themselves from other organizations by expanding both in size and number of new locations, thus creating new opportunities in diverse geographic areas [1] and encouraging subsequent employment growth in their related industries [10]. Uber, Spotify and Airbnb, to name just a few, are examples of software startups that have grown rapidly. Their products are disrupting traditional markets and are putting well-established actors under pressure.

H. Edison (✉)
Lero, NUI Galway, Galway, Ireland
e-mail: henry.edison@nuigalway.ie

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_15

To compete in this age of disruption, large and established organizations cannot rely on traditional ways of advancement, which focus on cost efficiency, lead time reduction, or quality improvement [33]. Corporate management is now looking for new ways to keep their leading positions in a fast moving market, and to innovate like startups. With greater resource in-house, they hope that they can bring innovative products with new customer values to market as startups do.

To increase the chances of radical product innovation success, Ries [34] proposed a set of critical innovation activities in what he called the Lean startup method. To capture customer value, an entrepreneur starts a feedback loop that turns an idea into a product quickly and then learn whether the underneath business hypotheses are valid or not. As the result, an entrepreneur might pivot to pursue a new direction of the business or continue and scale the proven business model. Pivot is common to most startups. It could prevent a startup from bankruptcy if time between pivots is minimized.

Even though the Lean startup approach is originated in software startups, it has also gained interest from large companies as General Electric, 3M, Intuit, Tieto, Supercell, etc. [23, 26]. More and more large and established organizations adopted the Lean startup approach, hoping that it will help them to generate successful software product innovation. A survey on 170 corporate executives reveals that more than 80% of them are deploying some elements of Lean startup approach in the Research and Development (R&D) activities. For example, they build minimum viable product (MVP) and test to the customers before making any investment, iterate based on their feedback, and measure and learn based on the data.

Definition: Lean internal startups share the characteristics of internal corporate venture (ICV) structure and Lean startup process.

The other way to adopt Lean startup approach in large organization is to establish what I called as Lean internal startups. Lean internal startups share the characteristics of internal corporate venture (ICV) structure and Lean startup process. As an ICV, Lean internal startups are a separate and dedicated entity within the organization and have access to its resources. Teams are responsible from end to end, from ideation to commercialization of a new software product. Moreover, the new product is targeted at external users or customers. Lean internal startups aim for radical product innovation and its business model which is different with the existing ones. In addition, Lean internal startups adopt the core principles of Lean startup: validated learning, build-measure-learn and innovation accounting. This will make a difference between Lean internal startups and ICVs.

Based on the experience of five new product innovation projects, in this chapter I will explore some challenges and pitfalls which we met during the concrete execution of Lean internal startups, their consequences and which action mitigated them and what was the final outcome.

2 Lean Startup, Internal Corporate Ventures and Lean Internal Startups

This section provides a discussion of the key concepts of Lean startup, ICVs and Lean internal startups and their relationship.

2.1 Lean Startup Approach

Inspired by Lean manufacturing principles from Toyota, Ries [34] introduced a scientific approach to manage successful product innovation in an extreme situation, in which the problem and solution are unknown. The Lean startup approach is built upon the Customer Development Model [8] which consists of four steps: customer discovery, customer validation, customer creation, and organization building. The first two steps are concerned with identifying what customers value the most. The last two steps aim to create a market for the product and scale the business. The model teaches to focus on and scale something that has been proven to work.

Principles of Lean Startup [34]:

– Entrepreneurs are everywhere
– Entrepreneurship is management
– Validated learning
– Build-Measure-Learn
– Innovation accounting

Lean startup is a structured process to validate business hypotheses through an engineering approach. Gilb and Gilb [21] refer Lean startup approach as a more "extreme" agile approach than extreme programming (XP) or Scrum to manage system building processes. Agile seems able to prescribe on how to develop a working software faster, but is still unable to give answer what product should be developed [11]. Although agile also advocates to build the software iteratively, it only works when the problem is known to the stakeholders. Figure 1 presents the key processes of the Lean startup approach.

Lean startup has five principles [34], as follows:

– Entrepreneurs are everywhere. Anyone can be an entrepreneur without owning a business, either a student or an employee within a corporation.
– Entrepreneurship is management. Startup is not only about product development but also business development.
– Validated learning. Startups are not to build new product or generate money, but to build a sustainable business. Hence, entrepreneurs should run experiments frequently and validate the hypotheses about the business model.

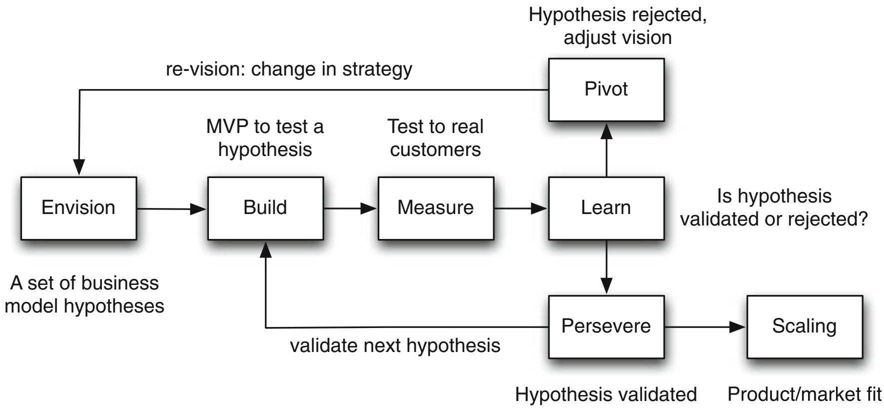

Fig. 1 Lean startup process steps [16]

- Build-Measure-Learn (BML). This is the fundamental activity of a startup: build an MVP, test it to market, measure customers' reaction and behavior, learn from it and use it to build a better product. An MVP is a smallest set of features or activities needed to test a hypothesis.
- Innovation accounting. To improve the outcomes, entrepreneurs must empirically measure and communicate the real progress of innovation.

Startups are not small versions of large organizations, as they are different in various aspects, e.g., goals, measurements, and culture [8]. For example, in terms of the goal, startups are to build a sustainable business model, while large organizations already have one. The first step is to identify the most riskiest business model hypothesis and then developing a minimum viable product (MVP) to begin the process of learning as quickly as possible. This requires actionable metrics that can demonstrate the cause and effect question. Based on the learning, entrepreneurs learn whether to persevere on the proposed business models or to pivot to a new direction, or to perish—renounce the business and the product [18, 34]. The key practices of Lean startup are summarized in Table 1.

2.2 Internal Corporate Ventures (ICVs)

In the literature, ICVs are commonly referred to as autonomous corporate startups [39] or internal startups [42] or innovation hubs [31] or spin-offs. ICVs are corporate entrepreneurial efforts that originate within a corporation and are intended from inception as new business for the corporation [24]. The introduction of a new internal venture may be the consequence of following or leading to product or market innovation [9, 38]. The degree of newness is defined by being new in the world and new in the industry [24]. New business can be established as an

Table 1 Key practices of Lean startup approach (adapted from [34])

Key practice	Description
Get-out-of-building	Confirm through face-to-face interaction with customers specifically what the problem is and whether it is worth solving. The purpose of early contact with customer is to understand the potential customers and their real problems
MVP	To validate the leap-of-faith assumptions, a version of product with minimum amount of effort should be released as quickly as possible. If MVP seems to have dangerous branding risk, launch MVP under different brand name
B-M-L loop	A feedback loop, which in order to turn ideas into products, measures how customers respond and learns whether to pivot or persevere
Use actionable metrics	Metrics that demonstrate clear cause and effect to evaluate the progress
Small batches	Engineers and designers work side by side on one feature at a time. Whenever that feature is ready to be tested with customers, they release to a small number of people
Pivot	Change in course or strategy. There are ten types of pivot proposed in [34]: zoom-in pivot, zoom-out pivot, customer segment pivot, customer need pivot, platform pivot, business architecture pivot, value capture pivot, engine of growth pivot, channel pivot, and technology pivot. A recent study [5] identified several new pivot types, including side-project pivot and complete pivot
Continuous deployment	The code written for an application is immediately deployed into production

instrument to pursue incremental innovation (a new product in a current market or a new market for a current product) or radical innovation (a new product for a new market). Innovation is generated through a separate and dedicated entity which is operated within an established company, using resources that are solely under the control of the company [30, 35].

The introduction of new internal ventures is also seen as a core process to create new capabilities for parent company [6, 28, 29]. ICVs need a great freedom to do experimentation [39, 41]. As a learning process, therefore, failure is inevitable. To minimize the cost and risk occurring with the entrepreneurial process, e.g., lack of track records, lack of familiarity with ICV's structure [9, 39], corporate management needs to balance between giving independency and monitoring [19].

Research on ICVs has no consensus on operation independence [19]. Some scholars advocate the structural separation of the venture from its parent. The bureaucracy and hierarchy that run corporation's main business do not support entrepreneurial effort [15]. Several empirical research find that the structural separation shows a positive impact on its performance (e.g., [7, 12]). Moreover, having this separate division with its own resources will not generate more impact to the companies [13]. On the other hand, other scholars argue that the parent company does not benefit from the separation. The nature of parent company could nurture the startup process, and at the end support the health of main business reciprocally

[29]. In addition, a quantitative study by Kuratko et al. [24] shows that its operation autonomy has no significant effect of ICV performance.

In the context of large software companies, a study by Raatikainen et al. [32] investigates how internal startups are used for new product development. The study finds that in each phase of a new product development life cycle, companies can apply different structures, e.g., internal startup, company subsidiary, incubating, etc. Another study by Selig et al. [36] investigates the role of corporate entrepreneurs in internal startup. The study finds that corporate entrepreneurs share the same characteristics as independent entrepreneurs. To further pursue innovation, corporate entrepreneurs need a guarantee of minimum salary and autonomy to experiment.

2.3 Lean Internal Startups

Current research on the Lean startup approach is centered on applying the approach in a software startup context to develop new product (e.g., [17, 20, 22, 27]). Very few studies investigate how the Lean startup approach supports software product innovation in large companies. In addition, research interest in applying Lean startup approach in non-startup context has emerged during the past years [40]. One of the Lean startup principles claims that entrepreneurs are everywhere, and that entrepreneurial spirits and approaches may be applied in any size company, in any sector or industry [34]. Therefore, applying startup concepts in non-startup contexts seems a promising avenue for established organizations to improve their innovation potential.

More and more large organizations such as General Electric, 3M, Intuit, Tieto, Supercell, etc. have shown their interest on Lean startup approach to leverage their product innovativeness [23, 26]. A survey on 170 corporate executives reveals that more than 80% of them are deploying some elements of Lean startup approach in the Research and Development (R&D) activities. For example, they build minimum viable product (MVP) and test to the customers before making any investment, iterate based on their feedback, and measure and learn based on the data.

Another way to adopt Lean startup approach in large organization is to establish Lean internal startup. Lean internal startups share the characteristics of ICV's structure and Lean startup process. As an ICV, Lean internal startups are a separate and dedicated entity within the organization and have access to its resources. Lean internal startups aim for radical product innovation and its business model which is different with the existing ones. Lean internal startups adopt the core principles of Lean startup: validated learning, build-measure-learn and innovation accounting, which differentiate them with traditional ICVs.

3 Cases Overview

3.1 Introduction

This chapter is based on five Lean internal startups, which were established in four different large organizations[1]: Lokki (F-Secure), FastCafé (CallBook), SeeSay (CallTech), SmallAds and CarAds (SellnBuy). The profiles of the case organizations and the teams are sufficiently diverse, which are illustrated in Tables 2 and 3.

To be qualified as Lean internal startups, projects need to show that they share the characteristics of ICV: separate and dedicated entity within an organization and have access to its resources, aim for radical innovation. Teams are responsible from end to end, from ideation to commercialization of a new software product. Moreover, the new product is targeted at external users or customers. In addition to this, projects also need to show that they adopt two or more key practises of Lean startup in their development process: validated learning, build-measure-learn, and innovation accounting.

3.2 Lean Internal Startup Activities

Lokki Case (F-Secure) Over the years, F-Secure has a long experience in various internal innovation initiatives, e.g., ground-up innovation, 10% free time, hackathons. In 2012, F-Secure was running a growth strategy exercise to explore a new opportunity area, which was people protection, develop a concrete product and release the product in Summer 2013. The top management decided to run the exercise inside F-Secure and called it an internal startup. It is different than Research and Development unit, as the business target has been set up for them.

To find a concrete idea, the team was brainstorming a number of family protection concepts and exploring consumer security products available on the market. Several product concepts were generated and validated through multiple online surveys to identify which concepts were more interesting to users. Then the survey was distributed through social media and the discussion boards in the USA and Europe where family-oriented people were discussing about kids and families. The survey results showed that there was a big interest for family location sharing applications.

The concept ideas were pitched to the top management in December 2012. Later on, it was decided in March 2013 that the key performance indicators (KPIs) of the internal startup were the schedule and net promoter score (NPS). The product

[1]Except Lokki (F-Secure), the name of the product and organizations are changed due to confidentiality reasons.

Table 2 Profiles of the four case organizations

	Lokki	FastCafé	SeeSay	SmallAds and CarAds
Organization name	F-Secure	CallBook	CallTech	SellnBuy
Domain	Cyber security and privacy	Print directory publisher	Telecommunication	Online classified advertisement
Total number of employees	>1000	>2000	>35,000	>450
Total revenue (million)	€ 146 (2015)	€352 (2015)	€14,000 (2015)	€163 (2015)

Table 3 Profiles of the five Lean internal startups

	Lokki	FastCafé	SeeSay	SmallAds	CarAds
Size	6–7	7–15	4–20	5–6	5–7
Composition	Team lead, architect, UX designer, 3–4 developers	Team lead, UX designer, 3 developers, 2 part-time members at start and now has 15 members	Team lead, 3 developers (interns) at start and now has 20 members	Team lead, 3 developers, part-time consultant, part-time UX designer	Team lead, UX designer, 3 developers, 2 part-time business developer
Product type	Family location sharing	Online prepayment platform	Audio and video conversation platform	Online payment solution	Online ads platform for used cars
Timeframe	2013–2014	2014–now	2013–now	2016–now	2014–2016

must be launched by mid of August 2013 and reach a certain of NPS by the end of September 2013.

Lokki 1.0, the first MVP was built using HyperText Markup Language (HTML) 5 technology because it was the standard technology for developing mobile applications. However, the result was not good enough because Lokki 1.0 had performance problems. Hence, in Lokki 2.0, the team pivoted to use native Android and iOS technology. Based on the Lokki experience, later on HTML5 is no longer a mandatory technology for mobile applications in the company.

The MVP was demoed to the management in May 2013. The main features were inviting users to a closed group and showing/hiding their locations. Then the team did a survey in some primary schools in Helsinki to test the product and gather feedback from parents and their children. The first version was released for iTunes and internal users in June 2013, and then for Google Play in August 2013, and finally for Windows Store in January 2014.

In Summer 2013, the organization's strategy was updated toward privacy. The people protection area was no longer inside the boundary of the strategy. A new emphasis became more prominent in the organization's strategy. Early 2014, the corporate management decided to release Lokki as an open source software project collaborated with two universities in Europe and the USA.

FastCafé (CallBook) CallBook is one of the leading marketing organizations and one of the largest print directory publishers. The initiative for new product development was part of the company strategy to shift from print directory business to digital business. Revenues at its print-based business are declining at an average 15% a year. In 2012, CallBook invested in innovation skills by bringing in a design consultant, to kickstart a design thinking ability for its new product.

The first MVP of FastCafé was using short message service (SMS) where people could send an SMS for ordering a coffee. The solution was tested with the people in the company to see how it could work. The team got around 20 orders in 1 week. To test with a broader audience, the second MVP was developed—an HTML-based mobile application using which customers could select the coffee, the café, and the pickup time. Once the order was received, the team would go to the café and place the actual order. As the customers came and picked up the coffee, the team interviewed them about their experience. The MVP was tested for 3 days. The team generated 700 dollars worth of orders and collaborated with 4 cafés.

After the meeting with the CEO, in October 2013, the team decided to focus on the development of FastCafé. The team prioritized to develop the order and payment processing. They worked together with the café owners to cocreate processes that fit into their current processes seamlessly. However, it was still unclear how the business model would be. At the first launch in February 2014, the team ran 1 month pilot program to see how the payment system could work. The pilot program collaborated with 20 cafés. It was extended for another 3 months for stress test of the system.

It took 10 weeks after its launch for the team to get the first revenue from the customers. Only a few customers signed up FastCafé and started paying after the pilot program. FastCafé was able to generate 20–30 dollars a week. The team learned that customers were willing to explore a new way of payment if it was convenient. For example, customers asked for PayPal to handle the online payment. However, the team considered that it was not financially viable as PayPal charged 4 dollars per transaction. Hence, the team had to find a new way to secure the payment process.

After the pilot program, the team started charging the cafés if they kept using FastCafé. The business model was the subscriptions model, which is based on the volume of transactions. The café would pay the weekly fee and a percentage of revenue earned through FastCafé. For electronic payment system, FastCafé used electronic funds transfer at point of sale (EFTPOS). The payment from customer is directly deposited to the café's bank account.

SeeSay (CallBook) SeeSay is an ongoing internal startup at CallTech, one of the leading telecommunication companies. CallTech is considered as a hierarchical and bureaucratic organization by the interviewees. Traditionally, as a telco, CallTech is providing a good infrastructure and technology for telecommunication network, including Internet connection. CallTech is looking for product innovation beyond the existing technology and launches an intrapreneurship initiative in-house. SeeSay was born internally in the initiative and established by one of the employees, who has now become the VP of SeeSay.

Over the years, CallTech has used to outsource any software development to external companies. Later on, there is an increasing understanding of having internal software development to seek an opportunity from the existing technology. It was the Team Lead vision to develop a new audio and video communication tool when he was working on another project. The existing video conference solutions were not able to solve their problems.

Three internship students were recruited to work on this project. In 2013, the first MVP was released and demoed internally. When the internship period was over, the development was taken by a group of internal engineers. The team spent a long time to establish a solid team. At the beginning, they got to know each other and find each role in the team. Everyone had a lot of ideas and wanted their idea to win. Hence, most of the time, they were discussing and figuring out what should be in the product. Once agreement was made about the roles, the team started being more productive.

All the users' feedback were put into the product backlog and the Team Lead prioritized and decided which stories would be implemented. The tasks were assigned by considering their scopes. However, each feature was implemented by the same person, from end to end. Every new feature was launched to the service and tested through an experimentation with real users. The team also evaluated existing features that had been in the service. Features that were no longer working anymore or least used by users were removed from the service.

At the time of investigation, the team was developing a premium feature, a paid version of SeeSay. The co-founder was taking responsibility to lead the development of premium features. At the same time, the size of the team was getting bigger. More new members with sales and marketing background joined the team to create a market for the product.

SmallAds (SellnBuy) SellnBuy is one of the largest online marketplaces. SellnBuy owns one of the largest newspapers nationally in the country. In product department there are 40–50 business developers who are responsible for managing different verticals, e.g., real estate and car. Each vertical has one or two business developers. Business developer searches for a business opportunity by talking to sellers (the ones who put ads on the online marketplace), understanding their needs and translating them into a new product idea.

The idea of SmallAds came as the result of competitor analysis in August 2015. As a business developer, the team lead realized that there was a trend that the number of professional sellers was decreasing. At the same time, a number of online marketplace companies, e.g., Amazon, Shopify, etc. have provided their own payment solution and APIs that could be used by customers to integrate and push their products into their marketplace. Hence, the sales is easily made in the same marketplace. In addition, the price to subscribe to this service is cheaper and the

API is easier to implement. There is a potential threat that their professional seller would switch to a different platform to increase their sales.

The concept of payment solution was presented to product director at the end of October 2015, and pitched to the CEO in November 2015. In February 2016, new presentation was conducted to the management team, but without any decision. The decision was made in April 2016. The approved business model is that the customers must pay a fee from every sales made through the platform. Since then, a formal recruitment was set up for internal employees. Some of the members were not fully dedicated to the team, for example, UX designers or consultants who had more knowledge about the existing product in the market.

The development was started in June 2016. The team were not developing from scratch but using component off the shelves (COTS) as much as they can. In July 2016, the agreement had been made for the front-end of SmallAds. The team started working on the back-end to allow the integration with partners' website. SmallAds will take care of all transactions made in the SellnBuy marketplace. Hence, customers would pay a small percentage of the sale. If they do not sell anything, then they do not pay anything. It is different with the existing business model in SellnBuy, where customers pay a subscription fee annually. The proposed business model was tested with existing users.

The first MVP was launched in October 2016. The main KPI of the team is how much sales are going through the new market place at the end of 2016. With this KPI, the main challenge to generate revenue is to convert users from existing verticals to the new marketplace. At the time of investigation, the team was preparing for the year end evaluation based on the KPI.

CarAds (SellnBuy) In 2014, corporate management wanted to explore a new way to sell and buy used vehicles for private people. Two employees worked on this project part time to explore this project. To identify the current problem with the car dealing, the team conducted user and customer analysis. With the help of an external consultant company, they interviewed 8 private car sellers and launched a survey to 1300 users of CallTech car vertical. The external consultant company selected respondents who bought or sold used cars in the last 6 months, either privately or through dealership. The results showed that even though there are high risk, insecurity, and hassle in the private car dealing, there is a willingness to pay for a solution that solves those problems.

To find a solution, the team looked at similar offers in the market and compared how these companies deliver customer value in the different steps of the selling and buying process. As the result, the team came up with new hypotheses about end-to-end car dealing process: the car will be sold within 60 days, otherwise it is bought by them.

It took half a year for them to identify the main problem and develop a product concept to solve that problem. The product concept was pitched to the corporate management. The management gave another half a year for the team to explore the possibility of having collaboration with car dealers. The concept was approved in mid-2015. On February 1, 2016, the full team was established and they got their own room to work on this product concept.

An experiment was set up to validate if the concept gained interest from the existing customers. An MVP was built to estimate the conversion rate they got in order to be more confident on what they should expect when it came to revenue and handling of the cars.

Even though the result was positive, the car dealers still did not believe in the concept and did not want to involve in the project anymore. They still considered the product as a threat. In June 2016, there was a new corporate strategy and the corporate management decided to focus the business further on the existing verticals and the project was terminated.

4 The Experience

This section discusses and summarizes the key challenges faced during the journey and some recommendations to address them.

4.1 Sponsorship

All cases show that Lean internal startup initiatives may come top down (driven by the top management) or bottom up (driven by the employees), or in other terms, as a planned initiative (Lokki, FastCafé, and CarAds) or an autonomous/opportunistic initiative (SeeSay and SmallAds) [24]. At their inceptions, the planned initiatives are considered as "strategically legitimate" since their initiations are purposeful and desirable. Their legitimacy increases the likelihood of the internal startups to receive resources from the company. On the other hand, for autonomous initiatives, the employees are responsible for initiating the Lean internal startup including the generation and implementation of ideas.

Challenge The case of Lokki, FastCafé, and CarAds reveals that since the organization strategy may change over time, it may also affect their legitimacy. Strategy and objective on innovation define how resources, product, and process are configured to support innovation [3]. As long as it can accommodate, the new strategy may not influence the legitimacy of the internal startups. However, the worst case happens when the internal startups are no longer within the new strategy. In such a situation, there is a high chance that the initiative will lead to a termination, despite whatever the results they bring.

The essence of top management support relates to effective decision-making to manage business risk that is outside the authority of the project team [43]. In the case of CarAds, the top management did not authorize the proposed business model, as it bore risk to the ongoing business. During the earlier phase of Lokki, the team had a champion sitting in the board of management. However, he did not have authority to influence his peer to protect the internal startup when there was a change in the organization's strategy.

In this critical time, to maintain the legitimacy, the initiatives should gain top management support or championship. The support from top management is crucial to protect the initiative, to mobilize resources, to empower and to inspire employees to innovate that whatever they are doing has impact on the organization [2]. As shown in the case of FastCafé, the support from CEO made a difference with other cases. Similar to Lokki, there was a change in strategic level as new management came in. However, the full support from the CEO enabled them to achieve their target and grow.

Lessons Learned The most obvious lesson is that top management support is critical to gain legitimacy. The team should be able to convince that the idea is important for the organization and then ask the legitimacy from the top management. From Lokki case, we can also learn that necessary changes should be made to ensure that the idea is still within the organization strategy.

Lean internal startups need to balance the need of their "customers": external customers and top management. Focusing on external customers enables them to build the right product, while focusing on top management secures their legitimacy in the organization.

4.2 Autonomy to Experiment and Pivot

In product innovation, where the target is the emerging or untapped market, traditional business and market analysis do not lead to profitability because of the inefficiency and inaccuracy of the business and market learning [4, 25]. Therefore, when Lean internal startups attempt to generate product innovation, they require to some extent freedom to learn through experimentation and gain new knowledge about products, technologies, markets, and organization routines [19].

Challenge Having autonomy to experiment and pivot means that Lean internal startups do not need to follow the existing rules in an organization. Some internal processes do not allow them to build and learn faster. In the case of Lokki, breaking the rules enabled them to speed up the process. They can choose any processes, tools, or platform that work for them. They do not depend on other units to solve particular problem. However, on the other side, in large organizations, people have already their own job descriptions. Breaking the rules is considered a violation to their territory and could raise an internal conflict among the employees themselves.

We observe a common pattern that Lean internal startups are not allowed to do business model-type of pivots. One of the reasons is, first, as shown in FastCafé during the envision phase, the idea that has been presented to and approved by top management has shown the biggest potential to generate revenue. Second, with the approved business model, the revenue has been set as one of the KPIs for the team, e.g., in the case of SmallAds. When the management evaluates that the product does not bring a significant value to the company, there is a chance that the startup would be terminated. Third, the top management sets out the boundaries for the team to experiment in business model, as shown in the Lokki case and echoed by CarAds.

Lessons Learned The adoption of a new method raises organizational tensions that play either constructive or destructive role. Tensions can lead to a detrimental effect on the organizational performance or be beneficial by fostering competition and challenges with management involved in the process of continually resolving tensions. Thus, management needs to find strategies to better manage and ensure that tensions align with organizational roles. A new culture and cooperative mode needs to be built for internal startups to work efficiently and coexist with existing business.

While innovation is critical for organizations to grow, it is also unpredictable and challenging to manage. Moreover, the essence of innovation is to find and learn about customers and their problems. Internal startups may fail to achieve their target, but they learn something. Acknowledge this and allow them to pivot, even in the business model.

Similar as external startups, the true product of Lean internal startups is the business model, not the solution itself. For large organizations, there may be a fear that the new product will hamper their existing business. However, there is no revenue without a business model. A practical tip to address these issues is not to have a risky business model or unique value proposition. It means that the new business model should not be targeted at the existing customer or market segment as it can adversely affect the overall performance of existing business.

Another tip is to have a clear mandate for the team on their mission and minimize the disturbance of work. Collaboration with various experts inside the organization is invaluable for the team; however, it should be done in a way that allows for faster development speed and faster learning.

4.3 Team Dynamics and Motivation

Based on the origin of the idea, startups can be divided into “idea-first startup” and “team-first startup” [37]. Our cases suggest that a planned internal startups share similar characteristics as team-first startups, while an autonomous internal startups as idea-first startups. In an autonomous initiative, the person who owns and develops the idea becomes the team lead. On the contrary, in a planned initiative, the team lead is selected from middle management. As compared to standalone startups, Lean internal startups enjoy security and less pressure.

Challenge In the case of Lokki, it was not easy to build an entrepreneurial team internally. It means that the team has core competence to build the product. Even when it was a planned initiative, not everyone in the organization is exciting about entrepreneurship. For example, it was not clear what happened if they failed to develop new product. Also, it was not clear what they would gain if it was a success.

Some people would prefer stability and predictability rather than working in an uncertainty.

This situation became worst in the case of idea-first startup such as SeeSay and SmallAds. The team lead had to recruit interns to work on the initial idea during the early phase. The idea of internal startup did not get interest from employees. In SeeSay case, the team lead had to compete with other teams to get resources, including the members.

Building a solid team takes time, especially if it is an idea-first startup where the idea is owned by the team lead. The vision should be shared among the members, so that they can work in harmony. In the case of SeeSay, they made an agreement about the roles and responsibilities and the development process. Moreover, in all cases, there was no incentive to get the startups succeed. As part of a large organization, they had a specific KPI as the basis for a pay raise or bonus. There was no additional reward given to the team.

Lessons Learned The general lesson in this case is to provide an incentive for people who are willing to initiate or join the internal startup so that each member has personal stake in the outcome. It is not necessarily related to financial benefits, such as bonus, pay raise, or stocks. As shown in the case of FastCafé, the decision to join the team was driven by intrinsic motivation rather than extrinsic motivation. Having new experience is the biggest motivation to involve in this innovation initiative. Some of the members had to demote from their positions before joining the team but working together with the experts seemed a good opportunity.

Having cross-functional team is necessary for Lean internal startups. It can be a mix of experts and fresh graduates. We can learn from the case of FastCafé, which the members were selected based on the skills and attitude. The team members were individuals who had deep knowledge in one or two areas but still had adequate knowledge across all areas more broadly so that they were able to interweave with other disciplines to fill in any gaps.

Employees are not liable, they are assets. Thus, investing in employees is always good to improve their capabilities and competence, which at the end will increase the likelihood of internal startup success. Moreover, trust your employees by giving them autonomy and freedom to be fast but expect responsibility.

5 Conclusion

Lean startup approach has gained interest from large and established organizations as a vehicle to pursue radical innovation. Ries [34] argues that the core ideas behind Lean startup can offer benefits for large and established companies as well. If the obstacles can be minimized, the opportunities can be very beneficial to support software product innovation. Lean internal startup is one way of Lean startup adoption in large and establish organizations, which is aiming for radical innovation.

The main challenges of Lean internal startups are due to the main characteristics of large organizations. They already have successful ongoing business in the market and they are backed up by strong bureaucracy, policies, and procedures to maintain the business. FastCafé is a testimonial of how large organizations can innovate like startups with full support from top management. SeeSay and SmallAds are examples of how large organizations nurture autonomous Lean internal startup in a controlled environment. From Lokki and CarAds we learn things to be considered carefully when deciding to run Lean internal startups. Even if the lessons we have drawn from these cases will need further validation and will certainly be improved or surpassed, we hope sharing them will contribute to further successes in similar endeavors.

Acknowledgements This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 754489.

References

1. Acs, Z.J., Mueller, P.: Employment effects of business dynamics: Mice, gazelles and elephants. Small Bus. Econ. **30**(1), 85–100 (2008)
2. Aiman-Smith, L., Goodrich, N., Roberts, D., Scinta, J.: Assessing your organization's potential for value innovation. Res. Technol. Manage. **48**(2), 37–42 (2005)
3. Akman, G., Yilmaz, C.: Innovative capability, innovation strategy and market orientation: an empirical analysis in Turkish software industry. Inter. J. Innov. Manage. **12**(1), 69–111 (2008)
4. Assink, M.: Inhibitors of disruptive innovation capability: a conceptual model. Eur. J. Innov. Manage. **9**(2), 215–233 (2006)
5. Bajwa, S., Wang, X., Nguyen Duc, A., Abrahamsson, P.: "Failures" to be celebrated: an analysis of major Pivots of software startups. Empir. Softw. Eng. **22**, 2373–2408 (2016). https://doi.org/10.1007/s10664-016-9458-0
6. Birkinshaw, J.: Entrepreneurship in multinational corporations: the characteristics of subsidiary initiatives. Strat. Manage. J. **18**(3), 207–229 (1997)
7. Birkinshaw, J., van Basten Batenburg, R., Murray, G.: Venturing to succeed. Bus. Strat. Rev. **13**(4), 10–17 (2002)
8. Blank, S.: The Four Steps to the Epiphany. K&S Ranch, Manhattan Beach (2013)
9. Block, Z., MacMillan, I.: Corporate Venturing: Creating New Business within the Firm. Harvard Business School, Boston (1993)
10. Bos, J.W.B., Stam, E.: Gazelles and industry growth: a study of young high-growth firms in the Netherlands. Ind. Corpor. Change **23**(1), 145–169 (2014)
11. Bosch, J., Olsson, H., Björk, J., Ljungblad, J.: The early stage software startup development model: a framework for operationalizing lean principles in software startups. In: International Conference on Lean Enterprise Software and Systems, pp. 1–15. Springer, Berlin (2013)
12. Burgers, J.H., Jansen, J.J.P., Bosch, F.A.J.V., Volberda, H.W.: Structural differentiation and corporate venturing: the moderating role of formal and informal integration mechanism. J. Bus. Ventur. **24**(3), 206–220 (2009)
13. Chasteen, L.: Intrapreneurship: What companies should do to develop new products. In: Proceedings of Engineering Management Conference. pp. 533–536 (2003)
14. Decker, R., Haltiwanger, J., Jarmin, R., Miranda, J.: The role of entrepreneurship in us job creation and economic dynamism. J. Econ. Perspect. **28**(3), 3–24 (2014)

15. Dess, G.G., Lumpkin, G.T., McKee, J.E.: Linking corporate entrepreneurship to strategy, structure, and process: suggested research directions. Entrep. Theory Pract. **233**(3), 85–103 (1999)
16. Edison, H.: A Conceptual framework of lean startup enabled internal corporate venture. In: International Conference on Product-Focused Software Process Improvement, pp. 607–613. Springer, Berlin (2015)
17. Efeoglu, A., Moller, C., Sérié, M.: Solution prototyping with design thinking–Social media for SAP store: A case study. In: Communications in Computer and Information Science. vol. 447, pp. 99–110. Springer, Berlin (2014)
18. Eisenmann, T., Ries, E., Dillard, S.: Hypothesis-driven entrepreneurship: The Lean Startup. In: Harvard Business Entrepreneurial Management Case No. 812-095 pp. 1–26 (2013)
19. Garrett, R.P., Covin, J.G.: Internal corporate venture operations independence and performance: a knowledge-based perspective and performance: a knowledge-based perspective. Enterp. Theory Pract. **39**(4), 763–790 (2015)
20. Ghezzi, A., Cavallo, A.: Agile business model innovation in digital entrepreneurship: lean startup approaches. J. Bus. Res. (2018). https://doi.org/10.1016/j.jbusres.2018.06.013
21. Gilb, T., Gilb, K.: "lean startup"–the most extreme agile method by far. Agile Rec. **9**, 53–54 (2012)
22. Haniotis, J.: Innovation Jams: Lessons in Agile product development–An experience report. In: Proceedings of the Agile Conference. pp. 223–229 (2011)
23. Kirsner, S.: The barriers big companies face when they try to act like lean startups. Harvard Bus. Rev. (2016). https://hbr.org/2016/08/the-barriers-big-companies-face-when-they-try-to-act-like-lean-startups
24. Kuratko, D.F., Hornsby, J.S., Covin, J.G.: Corporate venturing: insights from actual performance. Bus. Horiz. **52**(5), 459–467 (2009)
25. Lynn, G.S., Morone, J.P., Paulson, A.S.: Marketing and discontinuous innovation: the probe and learn process. California Manage. Rev. **38**(3), 8–37 (1996)
26. Marijarvi, J., Hokkanen, L., Komssi, M., Kiljander, H., Xu, Y., Raatikainen, M., Seppanen, P., Heininen, J., Koivulahti-Ojala, M., Helenius, M., Jarvinen, J.: The Cookbook for Successful Internal Startups. DIGILE and N4S (2016)
27. May, B.: Applying lean startup: An experience report–lean & lean UX by a UX Veteran: Lessons learned in creating & launching a complex consumer app. In: Proceedings of Agile Conference, pp. 141–147 (2012)
28. McGrath, R.G.: Advantage from adversity: learning from disappointment in internal corporate ventures. Bus. Ventur. **12**(5), 121–142 (1995)
29. McGrath, R.G., Keil, T., Tukiainen, T.: Extracting value from corporate venturing. Sloan Manage. Rev. **48**(1), 50–56 (2006)
30. Narayanan, V.K., Yang, Y., Zahra, S.A.: Corporate venturing and value creation: a review and proposed framework. Res. Policy **38**, 58–76 (2009)
31. O'Hare, J., Hansen, P.K., Turner, N., Dekoninck, E.: Innovation hubs: Why do these innovation superstars often die young. In: Proceedings of the 9th International Design Conference (2008)
32. Raatikainen, M., Komssi, M., Kiljander, H., Hokkanen, L., Märijärvi, J., Mohout, O.: Eight paths of innovations in a lean startup manner: A case study. In: Product-Focused Software Process Improvement (LNCS 10027), pp. 15–30. Springer, Berlin (2016)
33. Rejeb, H.B., Morel-Guimares, L., Boly, V., Assiélou, N.G.: Measuring innovation best practices: improvement of an innovation index integrating threshold and synergy effects. Technovation **28**, 838–854 (2008)
34. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Business. Crown Business, New York (2011)
35. Roberts, E.B., Berry, C.A.: Entering new business: selecting strategies for success. Sloan Manage. Rev. **26**(3), 3–17 (1985)
36. Selig, C.J., Stettina, C.J., Baltes, G.: The corporate entrepreneur: A driving force for strategic renewal and radical innovation in established companies. In: Proceedings of the 22nd ICE/IEEE International Technology Management Conference (2016)

37. Seppänen, P., Oivo, M., Liukkunen, K.: The initial team of a software startup, narrow-shouldered innovation and broad-shouldered implementation. In: Proceedings of 22nd International Conference on Engineering, Technology and Innovation/International Technology Management Conference (2016)
38. Sharma, P., Chrisman, J.J.: Toward a reconciliation of the definitional issues in the field of corporate entrepreneurship. Entrep. Theory Pract. **23**(3), 11–27 (1999)
39. Simon, M., Houghton, S.M., Gurney, J.: Succeeding at internal corporate venturing: roles needed to balance autonomy and control. J. Appl. Manage Stud. **8**(2), 145–158 (1999)
40. Unterkalmsteiner, M., Abrahamsson, P., Wang, X., Nguyen-Duc, A., Shah, S., Bajwa, S.S., Baltes, G.H., Conboy, K., Cullina, E., Dennehy, D., Edison, H., Fernandez-Sanchez, C., Garbajosa, J., Gorschek, T., Klotins, E., Hokkanen, L., Kon, F., Lunesu, I., Marchesi, M., Morgan, L., Oivo, M., Selig, C., Seppänen, P., Sweetman, R., Tyrväinen, P., Ungerer, C., Yagüe, A.: Software startups: a research agenda. e-Informatica software Eng. J. **10**(1), 89–123 (2016)
41. Weiss, L.A.: Start-up businesses: a comparison of performance. Sloan Manage. Rev. **23**(1), 37–53 (1981)
42. Yli-Huumo, J., Rissanen, T., Maglyas, A., Smolander, K., Sainio, L.M.: The relationship between business model experimentation and technical debt. In: Software Business, vol. 210, pp. 17–25. Springer, Berlin (2015)
43. Young, R., Jordan, E.: Top management support: mantra or necessity. Inter. J. Project Manage. **26**(7), 713–725 (2008)

Software Startup Education: Gamifying Growth Hacking

Kai-Kristian Kemell, Polina Feshchenko, Joonas Himmanen, Abrar Hossain, Furqan Jameel, Raffaele Luigi Puca, Teemu Vitikainen, Joni Kultanen, Juhani Risku, Johannes Impiö, Anssi Sorvisto, and Pekka Abrahamsson

Abstract Marketing is a vital activity for software startups as they seek high growth. A specific type of digital marketing, growth hacking, in particular has attracted a lot of attention in software startups. Growth hacking is about utilizing low-cost marketing practices and existing platforms to rapidly increase the user count of a service. Though topics related to growth hacking such as marketing on a general level have been extensively studied in the past, growth hacking has not seen much direct interest in the academia thus far. As a result, we currently have few tools to teach growth hacking in startup education. In this chapter, we present two board games intended to serve as an introduction to growth hacking.

Keywords Growth hacking · Startup education · Startup gamification · Software startup

1 Introduction

Though most companies are concerned with growth, for startups growth is typically far more vital than it is for more mature organizations. Strategies for growth are various. In terms of growing through user or customer acquisition, marketing is a key activity. Marketing strategies range from, e.g., digital viral marketing to

The definitive version of this chapter was published as a scientific paper in: Kemell, KK., Feshchenko, P., Himmanen, J., Hossain, A., Jameel, F., Puca, R.L., Vitikainen, T., Kultanen, J., Risku, J., Impiö, J., Sorvisto, A., Abrahamsson, P.: Software startup education: gamifying growth hacking. In IWSiB 2019 Proceedings of the 2nd ACM SIGSOFT International Workshop on Software-Intensive Business: Start-ups, Platforms, and Ecosystems, pp. 25–30, Tallinn, Estonia, August 26 (2019). https://doi.org/10.1145/3340481.3342734.

K.-K. Kemell (✉) · P. Feshchenko · J. Himmanen · A. Hossain · F. Jameel · R. L. Puca · T. Vitikainen · J. Kultanen · J. Risku · J. Impiö · A. Sorvisto · P. Abrahamsson
University of Jyväskylä, Jyväskylä, Finland
e-mail: kai-kristian.o.kemell@jyu.fi; pekka.abrahamsson@jyu.fi

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_16

traditional forms of display advertising done through television advertisements. As startups operate under a notable lack of resources this usually limits their marketing options compared to more mature businesses.

Growth hacking is a marketing strategy [1] that focuses on low-cost practices and using existing platforms in creative ways. This makes it well suited for startups, and indeed, growth hacking as a construct was originally discussed in relation to startups. Currently, few academic studies directly related to growth hacking exist, even if software startups are keenly studied by academics [2]. Though marketing is a long-standing area of research in economic disciplines, and search engine optimization (SEO) and other areas of research closely related to growth hacking have been extensively studied in the field of information technology, growth hacking has not been directly studied academically.

Indeed, one of the goals of the original version of this paper was to spark interest in growth hacking in the academia, while also presenting two board games for teaching growth hacking. In this chapter, *we present those two board games for teaching growth hacking*. These games can help those involved in teaching startup entrepreneurship, as well as aspiring startuppers. The games can be downloaded using the links provided at the end of this chapter.

2 What Is Growth Hacking?

The construct growth hacking was popularized by Sean Ellis [3] in his blog about startup marketing.[1] Growth hacking is a marketing strategy according to Herttua et al. [1]. As the name implies, it is about using various growth hacking techniques or practices to “hack” the growth of a company, often a startup. To this end, we consider growth hacking *a process of rapid experimentation across marketing funnel, sales segments and other areas of the business including the product development, to identify the most efficient ways to grow a business*.

In practice, growth hacking is technology oriented and relies on using technical practices, with one of the main tasks of a so-called growth hacker in fact being (software) development [1]. This, Herttua et al. [1] underline, is one of the main differences between growth hacking and other marketing strategies such as viral marketing or guerilla marketing.

In academic literature thus far, the following characteristics have been associated with growth hacking [1, 3, 22]:

- Use of data in the form of metrics
- Changing the service based on data
- Low-cost practices
- “Pulling” users to the service as opposed to

[1] www.startup-marketing.com

- "Pushing" the service to them
- Using existing platforms in creative ways
- A/B testing

Growth hacking is typically discussed by focusing on various growth hacking techniques often referred to as growth hacks [4–6]. Growth hacking techniques are numerous (see e.g. [4]). They range from social media practices such as following individuals or organizations in hopes of gaining followers in return to sales-related practices such as offering free software trials or downselling upon subscription cancelation. These practices are seldom exclusive and can be combined and experimented with at will by software startups.

A famous example of growth hacking in practice is the story of Hotmail. To tackle their growth issues early on, Hotmail implemented the signature text "PS. I Love You. Get your free e-mail at Hotmail" into all e-mails sent from their service. Having tried various other forms of marketing, this proved to be far more effective. Following the campaign, Hotmail quickly grew from a few thousand to a few million users and sold its service to Microsoft. In this fashion, growth hackers aim to "hack" growth by both being creative with existing platforms and using low-cost practices to drive growth.

3 Two Board Games for Learning or Teaching the Basics of Growth Hacking

In order to teach growth hacking in a fun and engaging way, we have developed two board games focused on growth hacking and various growth hacking techniques recommended by practitioners [4–19]. Aside from providing an overview of the categories of growth hacking techniques, the board games also offer practical examples of the use of individual growth hacking practices. Both of the games can be downloaded from FigShare, the link to which is in the final section.

One of the board games, the Growthopoly, focuses on teaching different types of growth hacking (e.g., social media marketing), taking on an overview approach. The second game, Game of Growth, then focuses on individual growth hacking techniques. The contents of the two games thus complement each other. Both games are available on FigShare via the following link: http://bit.ly/gh-board-games.

3.1 Growthopoly

Growthopoly (Fig. 1) is a Monopoly-inspired board game on growth hacking. In Growthopoly, the players compete against each other with the objective of gaining 5000 followers, and the player to reach that milestone first is the winner.

Fig. 1 The Growthopoly game board

At the beginning of the game, each player: (1) Chooses a player character; (2) Chooses an additional marker, in addition to their game character, for displaying the number of followers in the middle of the board; and (3) Receives a certain amount of game money. Money is expended (and gained) over the course of the game by landing in the various squares on the game board.

The player character is used as a game marker for moving on the board according to the die rolls (and other events). Each player character specializes in one of eight areas of specialty in growth hacking: (1) Search Engine Optimization, (2) Email Marketing, (3) Social Media Marketing, (4) Public Relations, (5) Product Development, (6) Display Advertising, (7) Content Marketing, and (8) Search Engine Marketing. Learning different growth hacking strategies is a central part of the game, akin to purchasing properties in monopoly, and these specialties help the character learn certain strategies faster.

The game then proceeds in turns, one player at a time. During their turn, the player throws two dice and advances the number of spaces denoted by the dice with their game character. What happens then depends on which type of space the player lands on. Sometimes the player has to act, while sometimes they can choose (not) to act. The game board contains six types of spaces:

- *Growth hacking skill space*. Whenever a player lands on a growth hacking skill space, the player may pay game money to study that skill for a number of turns: one turn for level one, two for level two, and three for level three skills. When the player has learned the skill, they gain the number of followers on the space.

- *Bonus space.* Upon arriving in a bonus space, the player draws a bonus card. Bonus cards are always positive and grant either money, followers, or both.
- *Trade fair space.* In this space, the player may pay a certain sum of money to gain a number of followers.
- *Problem and Solution (prob and solve) space.* The player draws a card, which may be either a problem or a solution. Solution cards are used to tackle problems and may be stored for later use, while problems cause immediate, negative effects when drawn unless countered with a solution. Players may trade solutions.
- *The Slush space.* The player spends a maximum of three turns at Slush. At the start of each turn, the player rolls a die to determine whether they stay or leave Slush.
- *The Start space.* Upon arriving in (or simply passing by) the start space after looping around the game board once, the player gains customers and game money.

By learning the different growth hacking techniques for their characters and by landing on the various spaces, players can gain more followers and/or more money. If a player lands on a growth hacking skill already learned (owned) by another player, the owner gains the number of followers listed on the space. If the player lands on the growth hacking skill that is also their player character's specialty, they get twice the number of followers and learning the skill takes one turn less than specified on the space. The game proceeds until one of the players reaches the goal of 5000 followers.

Growthopoly is intended to serve as a general introduction to growth hacking. It teaches the players about various types of growth hacking (e.g., Search Engine Optimization). It does not contain much educative content as far as microlevel growth hacking techniques go, however. The other game, which we present next, on the other hand focuses specifically on individual growth hacking techniques.

3.2 The Game of Growth

The Game of Growth (Fig. 2) is a cooperative board game. In the Game of Growth, the players form a team that is intended to emulate a startup organization. Rather than competing against each other, the team then aims to win the game together as a team while playing against the game board. The goal is to get 5000 followers for the team's hypothetical software service.

Before the game starts, the players choose what type of startup they want to be: tech, service, or entertainment. The team then starts the game with 5000 dollars in (game) money. Using the 5000 dollars, the players have 10 turns to reach 5000 followers. Each turn represents 1 week. Each turn has three phases, each of which is denoted by drawing a different type of card.

1. First, at the start of the turn, the team draws an event card. The event card applies special rules for that turn. For example, the event card might have a beneficial effect such as making hiring a new employee during that turn cheaper, or a negative one.

Fig. 2 Students playing the Game of Growth

2. Secondly, after the event card is drawn, the team draws three hack cards. Hack cards contain ways to increase the number of followers of the service. For example, a hack card may require the team to pay a few hundred dollars for a chance to gain a few hundred followers by rolling the die favorably. The team may either use or ignore the hack cards, but they are all discarded at the end of the turn either way.
3. Finally, at the end of the turn, the team reveals an employee card at the end of the turn. The team then either hires or refuses to hire the employee, concluding the turn. Any employees hired by the team will have to be paid a salary at the start of each turn until the end of the game, or until fired. The employees offer various ways for the team to gain more followers.

The game then continues in this fashion until the team either wins or loses. The game automatically concludes after ten turns have passed, at which point the team loses if they have not reached the 5000 follower milestone. Otherwise, the team either loses by running out of money on the way or gains 5000 followers before the time limit is reached.

The educational value of the game is mainly in the hack cards. Each hack card contains descriptions of individual growth hacks. The cards cover techniques such as asking Internet celebrities to promote your service or sending personalized emails to targeted prospects ("cold emailing") as a very early-stage startup looking to gain its first users or customers. The Game of Growth, in other words, takes on a more microlevel approach to teaching growth hacking, whereas Growthopoly focused on the big picture.

4 Methodology: How the Board Games Were Designed

The board games presented in this chapter were created during the course "TJTS5792 Advanced Lean Startups" in the University of Jyväskylä. The games were developed by two teams of Information Systems (IS) students, under the supervision of the teaching staff of the course. For creating the board games, we conducted a multivocal literature review on growth hacking prior to the start of the course.

The board games were based on the contents of various books written by startup experts, which we discovered by means of a multivocal literature review (i.e., a literature review that included both peer-reviewed academic literature and unreviewed "gray" literature such as books by practitioner experts). Due to the lack of academic research on growth hacking, we focused on gray literature. Moreover, we limited the literature reviewed to books. As this was part of a university course, we wished to provide the students with clear reading materials.

The results were filtered based on the context "growth hacking" was discussed in in the books, using researcher judgment. The focus was on confirming that the book discussed growth hacking themes (user acquisition by means that could be considered growth hacking). After confirming relevance, we performed quality appraisal by checking the reviews of each book on either Good Reads or Amazon Reviews. Only books rated 3.5 or higher (on a scale of 0–5) were included into the reading materials.

The books were then read by the students during the first week of the aforementioned course, one per student. Afterward, the students presented the contents of their book to the other students in the course during the following lecture. After this, the students split into two teams, each tasked with creating a board game they felt taught the most important things about growth hacking, based on the readings. The creation process was supervised by the course staff who provided assistance when (if) needed.

Once the games were completed, the course participants and the teaching staff played both games during the third lecture. Though the games were not formally evaluated, the game session was considered enjoyable by the participants. The games were revised and further improved based on the feedback from the session. Afterward, the students utilized the content of the games (the categories of growth hacking from the Growthopoly, as well as any techniques from the Game of Growth where applicable) to split into groups and utilize growth hacking techniques from these categories.

Using student-created content in this fashion is not a novel discourse in the field of scientific education. It has increased in recent years due to Wiki-technologies [20]. Moreover, our recent empirical, scientifically reported experiences from student created board games in SE education have also been positive [21].

5 Conclusions: Managerial Implications

Growth is vital for startups and marketing is key in achieving it. Growth hacking is a type of digital marketing that focuses on innovative, low-cost practices, and is generally discussed primarily in relation to startups. Little academic research into growth hacking currently exists, although marketing in general and fields such as Search Engine Optimization closely related to growth hacking have been extensively studied in the past. We therefore presented two board games for teaching growth hacking.

The two board games presented in this chapter should primarily be of interest to those who teach startup entrepreneurship or work in startup ecosystems, as well as new startuppers. The games serve as an introduction into growth hacking. However, for those seeking to learn more about growth hacking techniques, we recommend the books used as source material for these games [4–19].

For educators, we also underline the educational value of having had the students utilize the growth hacking practices in a real setting. Based on our experiences, utilizing growth hacking techniques in practice resulted in lessons learned not covered in the books. For example, during a course on growth hacking, students who used growth hacking learned that (1) if the platform sells display advertising by view count, views by bots also eat up the count. (2) Therefore, limiting ads to certain geographic locations can help not only to reach the right audience but also to avoid bots. (3) When advertising, e.g., a YouTube channel, it can be beneficial to have the traffic pass through a redirect link in order to collect more data about the incoming traffic.

Key Takeaways

✓ Game of Growth can be useful for teaching (or learning about) growth hacking techniques.
✓ Growthopoly can be useful for teaching (or learning about) growth hacking strategies.
✓ Both games can be downloaded from: **http://bit.ly/gh-board-games**

References

1. Herttua, T., Jakob, E., Nave, S., Gupta, R., Zylka, M.P.: Growth hacking: exploring the meaning of an internet-born digital marketing buzzword. In: Zylka, M., Fuehres, H., Fronzetti Colladon, A., Gloor, P. (eds.) Designing Networks for Innovation and Improvisation, pp. 151–161. Springer, Cham (2016)
2. Unterkalmsteiner, M., et al.: Software startups – a research agenda. e-Informatica. Softw. Eng. J. **10**(1), 89–123 (2016)
3. Geru, M., Rusu, E., Capatina, A.: Growth hacking practices in a start-up: a case study on thecon.ro. In: Proceedings of the 2014 International Conference on Risk in Contemporary Economy (2014)

4. Happy, A.: How I Create Growth Hacking Plans for Startups for $10,000: + TOP 300 Growth Hacks You Can Put into Practice Right Away. CreateSpace Independent Publishing Platform, Scotts Valley (2016)
5. Patel, S., Wormley, R.: 100 Days of Growth Book – 100 Actionable Tips to Grow Your Startup Faster. E-Book
6. Peters, R.: Growth Hacking Techniques, Disruptive Technology – How 40 Companies Made It BIG – Online Growth Hacker Marketing Strategy. Blep, Malden (2014)
7. Berger, J.: Contagious – Why Things Catch On. Simon & Schuster, New York (2013)
8. Berger, J.: How Ideas Spread. The Teaching Company, Chantilly (2014)
9. Berger, J.: The Hidden Forces that Shape Behavior. Simon & Schuster, New York (2016)
10. Covel, S.: Marketing Your Startup – The Inc. Guide to Getting Customers, Gaining Traction, and Growing Your Business. AMACOM, New York (2018)
11. Ellis, S., Brown, M.: Hacking Growth: How Today's Fastest-Growing Companies Drive Breakout Success. Crown Business, New York (2017)
12. Eyal, N., Hoover, R.: Hooked – How to Build Habit-Forming Products. Penguin, New York (2014)
13. Fong, R., Riddersen, C.: Growth Hacking: Silicon Valley's Best Kept Secret. Lioncrest (2017)
14. Fried, J., Heinemeier-Hansson, D.: Rework. Crown Business, New York (2010)
15. Holiday, R.: Growth Hacker Marketing – A Primer on the Future of PR, Marketing and Advertising. Portfolio, New York (2013)
16. Linkner, J.: Hacking Innovation. The New Growth Model from the Sinister World of Hackers. Fastpencil, Campbell (2017)
17. Snow, S.: Smartcuts – How Hackers, Innovators, and Icons Accelerate Success. Harper Business, New York (2014)
18. Walsh, R.: The Web Startup Success Guide. Apress, Berlin (2009)
19. Weinberg, W., Mares, J.: Traction – A Startup Guide to Getting Customers. S-curves, Los Angeles (2014)
20. Wheeler, S., Yeomans, P., Wheeler, D.: The good, the bad, and the wiki: evaluating student-generated content for collaborative learning. Br. J. Educ. Technol. **39**(6), 987–995 (2008)
21. Kemell, K.K., Risku, J., Evensen, A., Abrahamsson, P., Dahl, A.M., Grytten, L.H., Jedryszek, A., Rostrup, P., Nguyen-Duc, A.: Gamifying the escape from the engineering method prison – an innovative board game to teach the essence theory to future project managers and software engineers. In: Proceedings of the 2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC) (2018). https://doi.org/10.1109/ICE.2018.8436340
22. Kemell, K.K., Wang, X., Ngueyn-Duc, A., Grendus, J., Tuunanen, T., Abrahamsson, P.: 100+ Metrics for software startups – a multi-vocal literature review. In: Proceedings of the 1st Software-intensive Business Workshop on Start-ups, Platforms and Ecosystems (SiBW 2018), Espoo, December 3rd, 2018 (2018)

Part V
Startup Stories Worldwide

Key Influencing Factors in Early-Stage Hardware Startups: A Trilateral Model of Speed, Resource, and Quality

Vebjørn Berg, Jørgen Birkeland, Anh Nguyen-Duc, Khan Khalid, Ilias Pappas, and Letizia Jaccheri

Abstract Hardware startups, i.e., wearable devices, robotics, and Internet of Things, are a significant sector of technology startups, in which software development is relevant and needed. Compared to pure software startups, hardware development in startup contexts lacks a systematic approach and guidelines. This chapter describes an empirical model that captures common elements in product development approaches among many hardware startups. Grounded from insights of 18 active hardware startups, we constructed a trilateral model of resource, speed, and quality in early-stage product development. For startups with relevant contexts, the work suggests the preparation of internal, external resources, and good practices of prototyping and matching prototypes to business activities.

Keywords Hardware startups · Trilateral model · Speed–quality trade-off · Software–hardware integration · Prototyping · Hardware entrepreneurship

V. Berg · J. Birkeland · I. Pappas · L. Jaccheri
Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway

A. Nguyen-Duc (✉)
Department of Business and IT, University of Southeast Norway, Bø i Telemark, Norway
e-mail: angu@usn.no

K. Khalid
Karachi Institute of Economics and Technology, Karachi, Pakistan
e-mail: khalid.khan@pafkiet.edu.pk

I. Pappas
Department of Information Systems, University of Agder, Kristiansand, Norway
e-mail: ilias.pappas@uia.no

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_17

1 Introduction

The barriers to starting a hardware company have never been lower, a result of the rising hardware ecosystem. Technological advances, rapid prototyping, decreased component costs, small-batch manufacturing, and fundraising platforms have renewed the interest for hardware startups [1]. With the Industry 4.0 revolution [2], the adoption and development of hardware-related technologies, for instance, the Internet-of-Things (IoT), cyber-physical systems, and robotics are becoming mainstream. According to Gartner's hype cycle, by 2020, hardware technologies will be in 95% of electronics for new product designs. The number of connected hardware devices available in the worldwide market is approaching 15 billion devices [3].

In July 2017, hardware device maker Jawbone became one of the most spectacular failures in the history of startups [4]. Jawbone is known for wireless technology, i.e., Bluetooth headset, wireless speaker, and lately fitness tracker. Despite grabbing almost 1 billion dollars during 10 years of its life span, Jawbone failed to hold on to significant market share for its product line. Their final key product, UP3 fitness tracker bands, faced various production problems, insufficient customer satisfaction, and stiff competitions from Fitbit and Apple in the same market. In 2016, the company stopped making and then selling their fitness trackers before eventually selling off their remaining inventory to a reseller [4]. Soon after, Jawbone discontinued its relationship with its outside customer service agency.

Startups of any kind are a challenge, and startups wherein hardware is involved are particularly challenging. Hardware startup context poses several challenges to traditional product development and innovation methods [5]. With recent advances in hardware prototyping (i.e., 3D printing and hardware development kits), the development of hardware-related products can be more agile and iterative [6]. For instance, a higher development pace and greater flexibility are reported to be facilitated by using Agile methodologies in hardware projects at Ericsson [3]. However, in general, the complex nature of hardware products imposes many dependencies and constraints such as competitors, vendor, platform, and competence dependency [7].

The development of a hardware product is difficult, as it consists of complex processes. It requires a merger of various types of hardware components with versatile software products and processes. Due to this complexity, it is really difficult to expect someone to get hold of all the issues and hence a set of guidelines would shed light on common steps, practices, and pitfalls during early stage of hardware startups. These guidelines should be simple but effective, for example, focusing on the delivery of minimal viable product as compared to developing a comprehensive product can help the startup in capturing the market share. In this chapter, we will not only introduce a new hardware startup model but will also propose recommendations for early-stage startups.

2 Hardware Product Development

Hardware startups are those startups that develop products with mixed hardware and software parts, including embedded systems, sensor devices, and advanced robotics [1, 2, 7]. As stated earlier as well as depicted in Fig. 1, hardware product is a merger of various software and hardware components. At the bottom end lie the actual hardware components which are the core of the product. These are small devices that hold the working of all other hardware components at one place. The sensors are used to get the input from the environment and systems-on-chip is the new dimension of information processing. Connectors are used for the information flow. On top of the hardware components, we need a middleware that would provide a layer of abstraction for wrapper applications such as cloud and mobile applications. These applications provide the user interface to the users and provide accessibility and ease of understanding.

Hardware product development is distinct from software development, as they need to handle hardware design and development, and manufacturing in addition to software development. They also have to deal with production and logistics issues

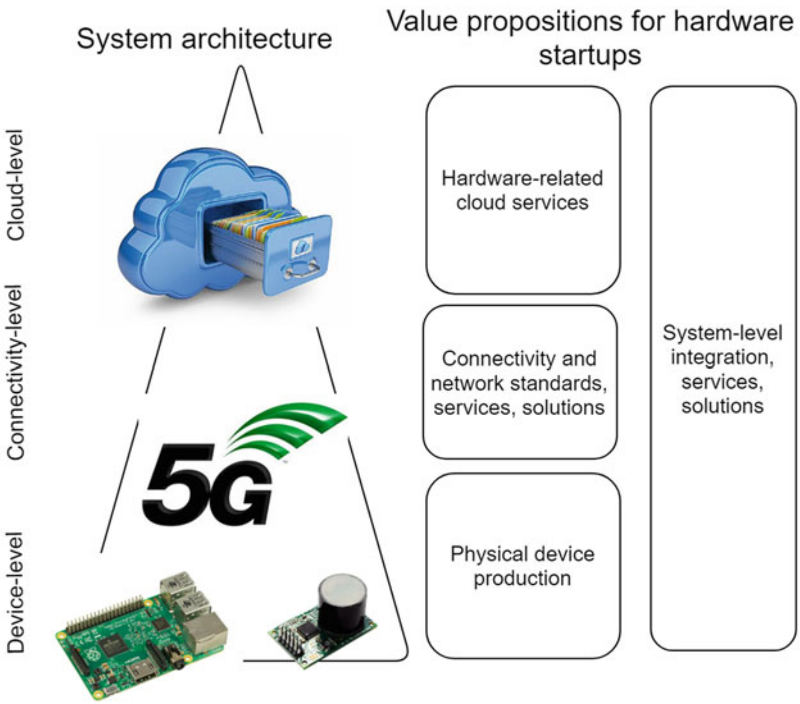

Fig. 1 Elements of startup hardware product

Table 1 Similarities among hardware and software development

<table>
<tr><th></th><th>Software development</th><th>Hardware development</th></tr>
<tr><td>Similarity</td><td colspan="2">Software relevant. Software is normally included in a hardware product
User interaction with products in various ways
Including design, implementation, testing, deployment, and maintenance
Considering both functional and nonfunctional requirements</td></tr>
<tr><td>Difference</td><td>Tangible product
Software is easier to change
refactoring and extension is common after a release
Some nonfunctional requirements can be relaxed
Various way of implementing software</td><td>Both tangible and non-tangible
Cost of change is much more
Physical component cannot be refactored
Nonfunctional requirements are critical
Expected significant design upfront
Heavy testing in various operational environment is expected</td></tr>
</table>

like packaging, shipping, and customs [1]. Table 1 summarizes key differences between a typical software development and hardware development process.

Even though it is a popular perception that hardware systems are associated with waterfall development methodologies, research has shown several positive experiences using Agile methodologies in embedded systems [5, 8–10]. Greene reported a positive experience of applying Agile approaches in firmware development at Intel [8]. Adopting XP practices, Santos et al. showed a successful software version created for control of a satellite camera [9]. Kaisti et al. conducted a systematic mapping study about the adoption of Agile methodology in embedded systems development [10]. The authors suggested that Agile practices can be used in the embedded domain, but the practices need to be adapted to fit the strictly constrained field of embedded system development. Compared to Agile software projects, some adjustment might need to be implemented, as shown by Ronkainen et al. when dealing with challenges of hard real-time requirements, prototyping, documentation, and test-driven development in Agile hardware development [5].

3 Study Method

3.1 Theoretical Framework

Researchers have proposed many software startup frameworks in the literature such as authors have discussed the issue of software process improvement in the context of achieving specific goals such as development speed, production cost, and product quality by using Grounded Theory approach [11], whereas a discussion of the current era of the digital economy is carried out in [12]. The paper states that companies are expected to develop high-quality software systems at "Internet speed" which means that the customer will get continuous value at a faster pace.

Work has been carried out in the direction of behavioral science and a framework is developed that has identified an inconsistency between managerial strategies and execution that can lead to failure. It was found that it is more important to understand the problem/solution fit before verifying product/market fit [13]. Authors have also discussed that the actual implementation of well-known practices such as Agile and Lean are not easy in a software startup because of the vagueness and imprecise implementation steps [14].

Early-age startups are counted as a separate discipline and Greenfield Startup Model (GSM) captures the underlying phenomenon of software development in early-stage startups [15]. The paper argues that product restructuring in light of customer requirement is more important as compared to quick product delivery to the market. The model illustrates that quick development is important due to a severe lack of resources, where low attention to quality leads to the accumulation of technical debt. The initial growth hinders the performance and future growth of the company. We selected GSM model as a starting point because of two reasons: first is the holistic coverage of almost all the process areas offered by the GSM model and second is its focus of the early stage of startups which is also our concern in this chapter.

3.2 Data Collection and Analysis

The study is of exploratory nature as we seek to create knowledge by investigating events and actions of those who experience them [16]. Semi-structured interviews of selected participants fitted both the time constraints and availability of hardware startups and are considered suitable for qualitative data analysis [16]. Interviews allowed for a discoverable approach, as interviewees could express themselves more freely and provide their own perspectives on personal experiences related to the research topics. Before the interviews, we looked into the cases' business background, through either their company websites or other relevant incubator or accelerator websites. Additionally, most participants answered a simple questionnaire prior to interviews where they pulled out basic information about themselves and the company. These measures allowed for efficient interviews as the first and second authors possessed more knowledge about the case and could use less time on initial formalities. Initial company analysis allowed for a holistic understanding of each case and provided stronger evidence for the conclusions drawn from the interviews. We applied the thematic synthesis process, which is a codes-to-theory model for qualitative research [17]. The first step of the analysis process was to read through the transcribed interviews (61 A4 pages of text) to generate initial ideas and identify possible patterns in the data. All interviews were transcribed shortly after they were conducted to ensure the actual meaning of interviewees' answers.

To validate the model's compliance with the hardware startup context, we performed a validation with the participating startups. We selected five Pakistani startups for model validation and each participant received the model with the

Table 2 Case demographics

Case ID	Product	Year	# people	Country	Current stage	Role
Case 1	Smart Gloves	2016	18	Norway	Early stage	C
Case 2	Medtech Biosensor	2017	5	Norway	Early stage	C
Case 3	Physical Exercise Game	2016	5	Norway	Later stage	C
Case 4	Unmmaned Aircraft System	2016	7	Norway	Early stage	C
Case 5	Advanced Noise Cancellation	2017	5	Norway	Early stage	C
Case 6	Medtech Hydration Monitoring	2016	10	Norway	Early stage	C
Case 7	LPG Management System	2016	8	Norway	Early stage	C
Case 8	Cable Cam System	2016	10	Norway	Early stage	C
Case 9	Digital Piggy Bank	2017	5	Norway	Early stage	C
Case 10	Collaborative Camera	2014	50	Norway	Later stage	C
Case 11	Interactive Children's Toy	2015	8	Netherlands	Early stage	C
Case 12	3D Printer Board	2009	1	Norway	Early stage	C
Case 13	Sensors for IoT	2007	25	Italy	Later stage	C
Case 14	Clique	2015	8	Pakistan	Later stage	V
Case 15	Electroid	2015	14	Pakistan	Early stage	V
Case 16	Eye Automate	2016	10	Pakistan	Later stage	V
Case 17	Car Chabi	2016	18	Pakistan	Later stage	V
Case 18	Xgear	2014	15	Pakistan	Early stage	V

Notation: In Role column, C means Construction and V means Validation

corresponding description and was asked to explain to what extent they found the model useful, and how the model contributed to describing the hardware startup context.

3.3 Case Description

Our case description, from Case 1 to Case 18, is summarized in Table 2. Our sample consists of startups in Norway, the Netherlands, Italy, and Pakistan. Most of the companies have less than 5 years of operation, are composed of 5–25 people, have launched their products, and acquired certain sale volumes.

4 The Trilateral Hardware Startup Model

Based on the thematic synthesis, we have created a model describing the context and overall engineering approach of hardware startups. The interview base of 61 pages of text allowed us to identify three higher order themes unique to hardware startups, before compiling the Trilateral Hardware Startup Model constituting a total of nine thematic elements.

4.1 Model Overview

From the collected data, we identified connections to quality, speed, and resources, operating as core elements of the trilateral model. Depending on the objective of a project, quality, speed, and resources are factors influencing product development. There may exist other elements that can contribute to describing the engineering activities in hardware startups (i.e., the scope of work); however, we found resources, quality, and speed to be the most prominent to the startup context.

As shown in Fig. 2, resources is the major element affecting hardware startups' ability to achieve both rapid development and high product quality. Experienced by most startups is that resources are restricted in most areas of the business, be it time, money, team capabilities, etc. For software startups, lack of resources is the most significant factor operating their context [18]. Limited access to resources set restrictions and boundaries to product development in both software and hardware startups.

Operating in competitive business environments characterized by extreme uncertainty, hardware startups must strive to obtain speed. Through evolutionary proto-

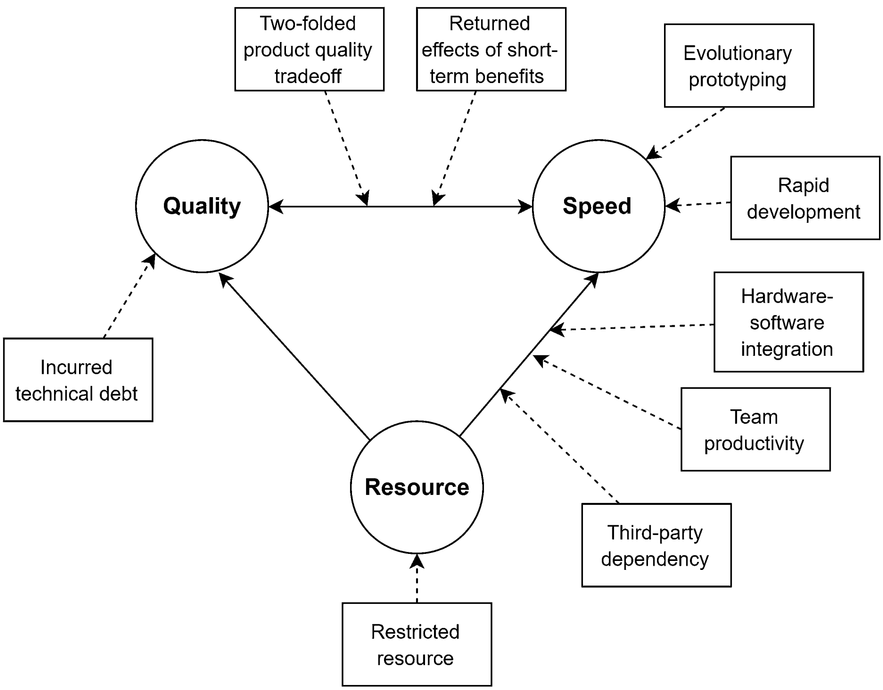

Fig. 2 The trilateral model of speed, resource, and quality in early-stage hardware startups

typing, rapid development, and simplified solutions, they continuously experiment to identify markets and customers. The importance of speed in startups has been emphasized by The Lean Startup method where the main objective is to grow the business with maximum acceleration [19]. In pursuing development speed, product quality tends to be less prioritized.

In the model, speed and quality are connected by a two-way arrow, illustrating a trade-off. Hardware startups are to a larger extent dependent on the quality of their products. Speed is achieved through simplified solutions on the software side while spending more time on the quality of hardware. The nine higher order themes can be seen as factors operating the hardware startup context. They impact each of the three elements or relations between them in various ways and extent. The following subsections will introduce each of the factors in greater detail.

4.2 Model Detail

4.2.1 Restricted Resources

Startup companies typically have a general lack of human, physical, and economic resources in their early stages [18]. These factors imply several constraints both as to how they manage and aim to grow their business and how they develop their products in extremely uncertain and dynamic market conditions [20]. Hardware startups experience similar restricted resources, evermore severe and harmful than software startups. Since they rarely have the capacity to develop prototypes themselves, they greatly depend on third parties to deliver ready-made or customized components in time. The resource intensity and complexity of embedded systems development leave immense demands to the capability of development teams; however, there is restricted access to dedicated people with both technical and entrepreneurial skills. As shown in Fig. 3, most of the hardware startups compose of software, hardware, and business competence. However, the level of competence and readiness of the teams for the developed products vary.

Poor economic conditions make recruitment challenging to perform. Lack of financial resources can be more severe for hardware startups as they require more initial capital due to the costs associated with rapid prototyping of hardware. This includes material costs and manufacturing fees, as compared to software development which is mainly associated with labor costs [21]. Access to prototyping equipment can help hardware startups perform more rapid prototyping; however, their lack of capital and facilities (e.g., labs or proper testing environments) hampers their prototyping capacity.

The insufficiency of human resources, finance, technical elements, and tools are major barriers to the process of prototyping.

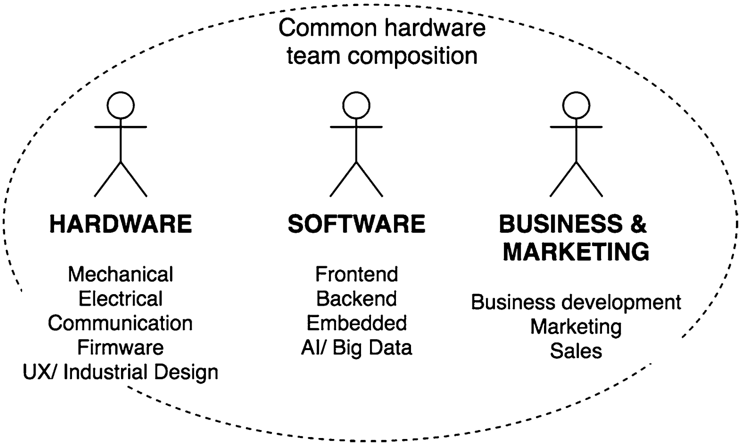

Fig. 3 A common set of human resources in hardware startups

4.2.2 Team Proactivity

The proactivity of the team significantly affects speed in the context of restricted resources. Anticipatory, change-oriented, and self-initiated team members are a necessity in the fast-changing, high-risk environment of startups. Hardware startups need team members dedicated to all aspects of the development process, including knowledge within the application domain, systematic development, software and hardware development, mechanical engineering, and experience of working with third-party companies. Working with several technology domains and external partners increases the complexity and lengthens the prototyping stage, and leaves higher demands to skillful teams and entrepreneurial capabilities. Members should have experience from working with bigger product companies as stringent financial resources and small production batches make it hard for startups to find manufacturers invested in the startups' success. Attracting experienced and knowledgeable people is hard as startups rarely can provide good salaries; hence, startups often consist of students with little or no experience from high-tech product development. Distributed development across big geographical distances and different cultures and languages can challenge the communication capabilities of the team. Communication is important to increase efficiency, reach goals, and avoid conflicts from misunderstandings. Since formal documentation practices imply documentation must be updated every time software is changed, documentation extensively relies on tacit knowledge to increasingly facilitate for rapid prototyping and implementation of new features. Effort and time can be spared when relying on the knowledge of team members.

Anticipatory, change-oriented, and self-initiated team members positively impact the prototyping process. Hardware startups demand more boundary spanning capabilities of their team members than what is required for software startups.

4.2.3 Twofolded Product Quality Trade-Off

Being able to prototype fast for testing new ideas and product features is crucial to learn faster than competitors. Business experimentation to build new features is performed in small iterative cycles with minimal effort on product quality to achieve speed. Since software can be changed and modified quickly, shortcuts and workarounds are more easily taken on the software side of hardware startups' products. Software testing is not systematically performed, rather the responsibility of each developer. While pursuing speed for the software development of the prototype, the nature of hardware development inhibits hardware startups in achieving rapid prototyping. The importance of nonfunctional requirements and the dependability on third parties are factors greatly affecting their ability to perform frequent releases. Whereas product quality has low priority in software startups, product quality in hardware startups can be seen as a twofolded trade-off between the quality of software and hardware. Hardware quality is often necessary to meet nonfunctional requirements and dependency toward third parties. Although the software and hardware contexts share many of the same characteristics, the specific hardware tasks and dependencies imply that speed sometimes is achieved through quality-adding activities. Shortcuts are mainly taken on the flexible software side. Hardware changes are kept to a minimum, hence requiring greater quality focus in early stages.

There is a need of synchronizing the quality level of software and hardware parts in hardware startups.

4.2.4 Third-Party Dependency

Without industry knowledge and the ability to mass-produce prototypes, hardware startups need external competence. Hardware startups depend on third parties for hardware production and physical components. Third-party dependency is the most prominent factor influencing product development in hardware startups. Long production and shipping times, manufacturing defects, end-of-life components, cost of rework, communication, and culture differences are some central issues affecting hardware startups' ability to perform rapid prototyping and business experimentation. Because of limited financial resources and small production batches, it can be difficult to find manufacturers invested in the startups' success. Working with local vendors producing components of high quality at an affordable cost is advantageous.

Long-term relationships with professional actors can enhance product quality and reduce the degree of dependencies. Access to prototyping equipment can reduce dependency on external partners, and have a positive effect on development time and prototyping costs, enabling faster problem space testing. Hardware development is a time-consuming process, and so minimizing necessary work for experimenting business ideas is essential.

Third-party dependency is not only an influencer in manufacturing, but also one of the most important factors determining the speed of hardware prototyping.

4.2.5 Hardware–Software Integration

Hardware startups' dependability on third parties for components and manufacturing inhibits both their flexible development approach and their ability to quickly develop prototypes. To achieve rapid prototyping for testing new ideas and product features, they facilitate changes in software. The software can be modified quickly according to changing customer demands, allowing for frequent releases. A flexible interface between hardware and software can enable more parallel and independent development of hardware and software and promote work on multiple solution methods. Another important asset provided by a flexible interface is the increased ability to handle product complexity, by allowing for parallel work and informal workflow. Documentation is to a large extent based on informal communication among the team members. The complex nature of hardware development and the numerous interaction points require information exchange to be explicit; however, too much documentation is not feasible in early development stages. Formal meetings consolidating efforts in hardware and software development are unavoidable, synchronizing and prioritizing new tasks.

4.2.6 Evolutionary Prototyping

Hardware startups usually start building a physical prototype to elicit requirements and achieve fast business experimentation. They extensively try to reuse software components, as physical components are easier to reuse with more refined prototypes. This is similar to an evolutionary approach, as they perform incremental improvements on an early low-resolution prototype. The approach can help them demonstrate problem/solution fit through discovering the needs of early customers, enhancing the effectiveness of the product, and prevent exhaustion of their severely limited resources. As in hardware development, we differentiate between works-like prototypes and looks-like prototypes. As seen in Table 3, the evolutionary approaches for these types are different. Regarding works-like prototypes, evolutionary prototyping promotes flexibility and reactiveness; however, hardware and hardware-related code are sensitive to frequent changes and refactoring due to

Table 3 Evolutionary prototyping approaches

	Works-like prototype	Looks-like prototypes
Examples	Ardunio+Raspberry Pi Replacement of wifi receivers to 5G receivers	Wireframes, foam, 3D printed layouts, etc.
Reuse approach	Reuse physical components Modular codes/components with clear interfaces	Reuse layout designs Clean, modifiable, extendable designs

the hard real-time requirements and third-party dependencies. Frequent changes might unconsciously change system behavior, and so rapid prototyping depends on predevelopment system design and planning activities beyond the purpose of the evolutionary approach.

There are different evolutionary approaches for works-like and looks-like prototypes.

4.2.7 Rapid Development

The most important priority of hardware startups is to achieve quick development speed. Testing and quality assurance practices are usually inferior to speed-related activities. The importance of testing new ideas and features on customers is crucial to learn faster than competitors, but achieving this in environments of severely restricted resources and third-party dependencies can be somewhat of a challenge. Hardware startups generally minimize any degree of process, preferring ad hoc development approaches customized to their own needs. They seek to utilize a small, flexible team without any bureaucracy, capable of responding quickly to changes. Informal communication and workflows are favored to formal documentation practices. Having a skilled, boundary-spanning team often counter-balances the lack of process. The same relates to testing practices which highly depend on individual efforts, manual smoke tests, and simulations in early phases. By following an evolutionary prototyping approach, hardware startups aim to test an initial prototype to elicit requirements. Problem space testing is a resource-intensive activity since each prototype involves individual production costs, emphasizing the importance of maximizing valuable learning from each. Having a professional local vendor can help decrease delivery times and manufacturing defects. Access to prototyping equipment can decrease development time and costs as third-party dependencies are reduced, further enhancing the ability to perform problem space testing. A flexible hardware–software interface increases the level of parallel and independent development and can help hardware startups better manage quality concerns in later stages. Since improvements and changes to software are quicker and easier, hardware startups should keep software development in-house.

Rapid development for hardware contains the balance of speed and quality.

4.2.8 Incurred Technical Debt

As hardware startups accept that time to market is a more important objective than product quality, development teams take shortcuts and workarounds. New features are implemented in small, iterative cycles to perform rapid business experimentation, with minimal effort on quality assurance and documentation practices. In some cases, this phenomenon occurs as "feature creeps," the excessive addition of new features for releases. While feature creeps could be as dangerous as it unnecessarily delays product launches and drives up costs, one common problem in both hardware and software startups is technical debts. Software features are implemented with a minimal amount of functionality. As the documentation would need to be updated for every change made to the code base, developers rely on their own knowledge instead of updating formal documentation. Since hardware startups rarely have the capacity to produce many prototypes, problem space testing becomes a challenging endeavor. The evolutionary approach increases the chance of feature creeps. Restricted resources and the need for rapid development lead to the accumulation of technical debt.

Compared to throwaway prototyping, evolutionary prototyping increases the chance of feature creeps.

4.2.9 Return Effects of Short-Term Benefits

With business growth as the main objective in the early phases, hardware startups will eventually have to slow down development to meet the ever-increasing needs of established customers. The evolutionary approach promoting reuse of components will lead to components holding too low quality or features not contributing to the core-delivered value of the product. The numerous interaction points between software and hardware components are vulnerable to later changes. Updating or removing code base can potentially change the entire system behavior, as failing to meet timing and performance constraints can jeopardize the system operation. This means that refactoring quickly becomes an immensely complex endeavor. Hardware startups favor informal communication and simple workflows. Shortcuts can speed up development in early phases, but might cause a severe amount of rework in the long run. In the worst case, lack of documentation and quality can put all development on hold. When scaling the business to a larger customer base, new employees are needed. Tacit knowledge makes it hard to integrate new people and can inhibit further growth. The introduction of more rigorous processes is necessary for the long run, but will forcibly deny the initial speed and flexibility of hardware startups.

Short-term benefits of rapid prototyping scale-up technical and organizational challenges in later phases.

5 Recommendations

Hardware startups develop a wide range of products, including systems that may cause critical situations if system operation fails. The findings in this chapter were synthesized from 18 case studies that share commonalities. These conditions could be thought as criteria to apply the recommendations from this chapter:

- Startups with bootstrap financing models, at their early stages
- Startup teams include both technical and business competence
- Startups operate more than 6 months, have actually launched or sold their products
- Startups whose business model lies in both hardware and software components

The study is designed and conducted to reduce as many validity threats as possible; hence, it should be valid to relate the findings in the trilateral model to hardware startups with the described criteria. The model explains the priorities of hardware startups in their engineering approach and provides a simple illustration of the hardware startup context. The model presents the specific needs for the process of managing the relationship between quality and speed under restricted resources. It can help practitioners obtain a better understanding and awareness of their own context. This is useful for hardware startups in understanding the underlying motivation for introducing specific practices and activities, and why some practices may counteract the overall objective of startups. The improved understanding will help practitioners in making technical and business-related decisions of sustainable character. Particularly, we suggest that:

(a) Find a hardware–software competence to become a co-founder. The core value proposition of your startup should be done in-house.
(b) Be aware of cost-effective external resources that are usable in the prototyping.
(c) Encourage proactive and boundary-spanning activities of team members during prototyping. All team members should involve in cross-domain activities, software–hardware, technology–business.
(d) Plan, store, and document for reuse components. The team stores and reuses designs of looks-like prototypes while enhances and reuses software/hardware components of works-like prototypes.

(e) Focus experimentation with core features, and stay away from feature creeps in early stages. Experiments on multiple features might delay the development process, reduce quality on core features, and deviate the business development.
(f) Align human resources and organizational structures according to the product structure.
(g) Plan for quality, not only speed. Some quality attributes require upfront specification, experimentation, and designs that are not able to add in during the later stages of prototyping.

6 Conclusions

Hardware startups develop physical products with mixed hardware and software components, requiring expertise within a broad range of technological fields. In addition to software development hardware startups deal with production and logistics issues, factors implying higher initial financial and human investments than what is experienced by software startups. With the trilateral model, this study explains the priorities of hardware startups in their engineering approach, providing a simple illustration of the hardware startup context. It presents the specific needs for the process of managing the relationship between quality and speed under restricted resources, providing practitioners with a better understanding and awareness of their own context. The improved understanding will help practitioners in making technical and business-related decisions of sustainable character.

For researchers, the trilateral model provides a first step toward understanding the context and engineering activities in hardware startups, outlining potential areas for future research. Future work should verify the model with other startup companies to find its applicability in other environments, enabling generalization to a larger startup audience. More investigations should be undertaken to understand the role of scope in the engineering activities of hardware startups, and how it can be included in the model. In addition, the three specific factors should be further explored in later studies. As hardware startups need more attention to hardware quality to allow for evolutionary prototyping and speed, there should be engineering approaches describing how hardware startups can manage the relationship between restricted resources and increased quality demands.

There are several limitations identified to this study. Having based our study on qualitative measures, results and implications are subject to bias. To mitigate the risk of misunderstandings or wrong interpretations, both researchers attended all interviews. Whenever possible, interviews were performed face to face on-site. Recordings were transcribed and translated shortly after each interview to ensure that respondents' meanings were preserved. Another limitation is the insufficient knowledge on technical decisions and product development challenges provided by some interviewees (i.e., knowledge of business executives is often based on managerial viewpoints). The results would benefit from a greater amount of participants providing insights into everyday engineering activities of hardware startups.

References

1. DiResta, R., Forrest, B., Vinyard, R.: The Hardware Startup: Building Your Product, Business, and Brand. O'Reilly Media, Newton, MA (2015)
2. Lasi, H., Fettke, P., Feld, T., Hoffmann, M.: Industry 4.0. Bus. Inf. Syst. Eng. **6**(4), 239–242 (2014)
3. Gustavsson, T., Rönnlund, P.: Agile adoption at Ericsson hardware product development. In: Proceedings of the 22nd NFF Nordic Academy of Management Conference, Reykjavik, Iceland, 1–23 August 2013
4. Juetten, M.: Failed Startups: Jawbone. https://www.forbes.com/sites/maryjuetten/2019/02/05/failed-startups-jawbone/ (n.d.). Accessed 31 July 2019 from Forbes
5. Ronkainen, J., Abrahamsson, P.: Software development under stringent hardware constraints: do agile methods have a chance? In: Marchesi, M., Succi, G. (eds.) Extreme Programming and Agile Processes in Software Engineering, pp. 73–79. Springer, Berlin (2003)
6. Larman, C., Basili, V.R.: Iterative and incremental developments. A brief history. Computer. **36**(6), 47–56 (2003). https://doi.org/10.1109/MC.2003.1204375
7. Nguyen-Duc, A., Weng, X., Abrahamsson, P.: A preliminary study of agility in business and production: cases of early-stage hardware startups. In: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 51:1–51:4. ACM, New York (2018). https://doi.org/10.1145/3239235.3267430
8. Greene, B.: Agile methods applied to embedded firmware development. In: Proceedings of the Agile Development Conference, Salt Lake City, UT, 22–26 June 2004, pp. 71–77 (2004)
9. Dos Santos, D., da Silva, I.N., Modugno, R., Pazelli, H., Castellar, A.: Software development using an agile approach for satellite camera ground support equipment. In: Elleithy, K. (ed.) Advances and Innovations in Systems, Computing Sciences and Software Engineering, pp. 71–76. Springer, Dordrecht (2007)
10. Kaisti, M., Rantala, V., Mujunen, T., Hyrynsalmi, S., Könnölä, K., Mäkilä, T., Lehtonen, T.: Agile methods for embedded systems development—a literature review and a mapping study. EURASIP J. Embed. Syst. 2013, 15 (2013). https://doi.org/10.1186/1687-3963-2013-15
11. Coleman, G., O'Connor, R.: Using grounded theory to understand software process improvement: a study of Irish software product companies. Inf. Softw. Technol. **49**(6), 654–667 (2007)
12. Baskerville, R., Ramesh, B., Levine, L., Pries-Heje, J., Slaughter, S.: Is "Internet-speed" software development different? IEEE Softw. **20**(6), 70–77 (2003)
13. Giardino, C., Wang, X., Abrahamsson, P.: Why early-stage software startups fail: a behavioral framework. In: International Conference of Software Business, 2014, June, pp. 27–41. Springer, Cham (2014)
14. Bosch, J., Olsson, H.H., Björk, J., Ljungblad, J.: The early stage software startup development model: a framework for operationalizing lean principles in software startups. In: International Conference on Lean Enterprise Software and Systems, 2013, December, pp. 1–15. Springer, Berlin (2013)
15. Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: the greenfield startup model. IEEE Trans. Softw. Eng. **42**(6), 585–604 (2016)
16. Oates, B.J.: Researching Information Systems and Computing. Sage, London (2005)

17. Cruzes, D.S., Dyba, T.: Recommended steps for thematic synthesis in software engineering. In: 2011 International Symposium on Empirical Software Engineering and Measurement, pp. 275–284 (2011). https://doi.org/10.1109/ESEM.2011.36
18. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: a systematic mapping study. Inf. Softw. Technol. **56**(10), 1200–1218 (2014). https://doi.org/10.1016/j.infsof.2014.04.014
19. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Constant Innovation to Create Radically Successful Businesses. Crown Books, New York (2011)
20. Coleman, G., O'Connor, R.: An investigation into software development process formation in software start-ups. J. Enterp. Inf. Manag. **21**(6), 633–648 (2008)
21. Wei, J.: State of the hardware incubators and accelerators in the United States [society news]. IEEE Consum. Electron. Mag. **6**(1), 22–23 (2017)

The Rise and Fall of a Database-as-a-Service Latvian Unicorn

Didzis Rutitis and Tatjana Volkova

Abstract This chapter presents a postmortem analysis of the collapse of a Latvian enterprise software company Clusterpoint that attempted to enter the global market with a Cloud-based database-as-a-service (DBaaS) offering to compete with MongoDB and other vendors in the NoSQL database category. The beginning of Clusterpoint is dated back to 2006 when three co-founders established the Clusterpoint Ltd. Company in Riga, Latvia. Clusterpoint developed its proprietary, closed source Clusterpoint database software and sold it in the local Latvian market using traditional enterprise licensing model. In 2015, Clusterpoint used more than 2 million EUR for launching its Cloud-based DBaaS offering and entering primarily the US market. In 2016, it was recognized by market research agency Gartner as one of Cool Vendors in platform-as-a-service (PaaS) segment. However, by the end of 2017 the company was unable to attract another investment round for financing its operations and was forced to file for insolvency due to liquidity issues. Research results suggest that the main reason for company collapse was related to the inability to achieve product/market fit due to the lack of Clusterpoint integration with the major Cloud infrastructure vendors (Amazon WS, Microsoft Azure, etc.) and its focus on closed-source business model. The effect of these two factors was amplified by premature scaling. The authors use the research method of an empirical case study and postmortem analysis approach, analyzing the outcome in the retrospect (doing the "research-in-the-past") of a completed project. One of the co-authors has worked as an employee at Clusterpoint during from February 2015 till March 2016.

Keywords Product/market fit · Pivot · Premature scaling · DBaaS

D. Rutitis (✉) · T. Volkova
BA School of Business and Finance, Riga, Latvia
e-mail: Didzis.Rutitis@ba.lv; Tatjana.Volkova@ba.lv

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_18

1 Introduction

The purpose of this chapter is to analyze the reasons behind the collapse of Clusterpoint company starting from 2015 when it attempted to enter US market with a Cloud-based database-as-a-service (DBaaS) using a postmortem analysis method.

1.1 Company Background

Clusterpoint was founded in Latvia on August 21, 2006 by three co-founders who knew each other from university and worked together at previous jobs. Initially, the company was developing an on-premises search engine, which was later transformed into a standard on-premises document database. The company gradually developed its proprietary, closed-source NoSQL database software and sold it in the local Latvian market using traditional enterprise licensing model only.

The company has not ever officially disclosed the total amount of raised venture capital. However, the public records and media releases provide information on the total amount of the raised capital of at least 5 million EUR since early 2008 until filing for insolvency in 2017. At least 2 million EUR were raised in 2015, including a bridge round from the existing investors [1, 2].

In March 2015, the company launched its database-as-a-service (DBaaS) offering, named Clusterpoint Cloud, and made it available over its corporate website—www.clusterpoint.com. The company management ambition was to enter initially in the US market due to NoSQL database popularity in this market and attract the first Fortune 500 enterprise level customer within the first year of company operations in the USA. Among biggest competitors in NoSQL category, there were NoSQL software vendors like MongoDB, Couchbase, IBM Cloudant, Amazon DynamoDB, and others [3].

Clusterpoint DBaaS service was hosted at Telia company data center in Riga, Latvia, and it was offered to end customers using subscription model where customers were charged using pay-as-you-go model only for the actually consumed amount of computational resources: CPU core time, Storage Space, RAM, and Outbound traffic, which was similar to Amazon WS and other Cloud-based database solution provider business models, who provided platform-as-a-service or hosted database services [4].

Key company's facts

- ✓ Founded in 2006 in Latvia
- ✓ NoSQL database vendor
- ✓ In 2015, launched database-as-a-service (DBaaS) offering Clusterpoint Cloud
- ✓ In total, raised over 5 million EUR of venture capital investment from 2008 to 2017
- ✓ Filed for insolvency in November 2017

1.2 Company Growth and the Consequent Challenges

In December 2014, Clusterpoint company shareholders hired a new CEO who brought his technical experience in work within distributed computing environment and NoSQL type of database solutions, as well as managerial experience from being a head of engineering team responsible for Google Web Search product development.

By the end of 2014 Clusterpoint had only ten employees, and the team was grown to 24 people by the end of 2015 [5]. By this time the company had managed to hire both experienced managers and technical engineers with outstanding professional experience from global corporations (e.g., Google, Intel, Oracle, PwC) and academic background from the global top universities (e.g., MIT, Yale), thus creating a team with a solid and promising background.

After spending nearly one million EUR of promotional budget in 2015 for Clusterpoint Cloud DBaaS marketing and generating its product awareness mainly in the US market using offline and online channels, the company did not succeed in bringing in any new business from abroad and converting approximately its free users into paying customers during 2015.

In the early 2016, Clusterpoint shareholders communicated to the company CEO that there will be no additional venture investment provided for marketing purposes in 2016 to continue the promotional and marketing activities for entering foreign markets and supporting DBaaS promotion. This was in contradiction to what was promised by shareholders to the CEO upon hiring him late 2014 when they promised to provide sufficient financing for marketing and promotional activities for at least 2–3 years long period. After receiving an update on such shareholder decision and feeling of the betrayal, the CEO decided by himself to step down in early 2016.

Soon after, a part of the remaining management team was laid off within several months. A new CEO was hired—an enterprise-level executive with previous experience in traditional telecommunications business from Australia, residing in London, UK, who hired a new team of foreign executives to support his activities for Clusterpoint Cloud commercialization and bringing in the first paying customers for Clusterpoint Cloud from abroad.

Unfortunately, the newly elected CEO and his team also did not succeed to bring in any new business from abroad, and the company suffered 1.24 million EUR losses at 0.43 million EUR turnover in 2016, which followed the 1.7 million EUR losses experienced in 2015 at 0.415 million EUR turnover from local operations and the enterprise sales in Latvia [5].

As a result, the newly elected CEO and his executive team stepped down by the end of 2016.

During 2017, Clusterpoint shareholders and management struggled to raise additional financing round but did not succeed and were forced to submit for insolvency in November 2017 [5]. The shareholder structure was not fully transparent as Latvian company has been fully 100% owned by UK-based Clusterpoint Group Limited, which also filed for insolvency in November 2017 [6]. Company House records reflect at least ten different private and legal persons, including original co-founders, as being Clusterpoint Group Limited shareholders throughout different periods of time [7].

2 Research Approach

2.1 Postmortem Analysis

The postmortem analysis (PMA) is an empirical study method in software engineering, commonly used for collecting historical data from completed projects where the outcome is analyzed in the retrospect. The aim of any PMA is to provide answers to the following four key questions in order to understand the possible improvements for the next development round [8]:

1. What did we do well that does not have to be further discussed (the positive aspects)?
2. What did we learn (the negative aspects)?
3. What should we do differently the next time (the negative aspects that require improvements)?
4. What still puzzles us (share the things that are confusing or do not make sense)?

In the context of this research paper, postmortem analysis approach and its key questions are used to describe the reasons behind the failure to achieve product/market fit by software vendor Clusterpoint in 2015 and subsequent years until filing for insolvency by the end of 2017. The second and third questions are combined to reflect those issues that likely caused the failure to achieve product/market fit in foreign markets for Clusterpoint Cloud DBaaS solution and should have been implemented differently.

The postmortem analysis was conducted by following the traditional four-step model (see Fig. 1) for small and medium enterprises (where the number of team members directly involved in the software development project ranged from 2 to 50) that includes [8]:

1. A project review to identify the most suitable methods for next steps
2. Objective and subjective data collection
3. Analysis of findings prioritized and synthesized as lessons learned
4. Publishing of the summary of the findings.

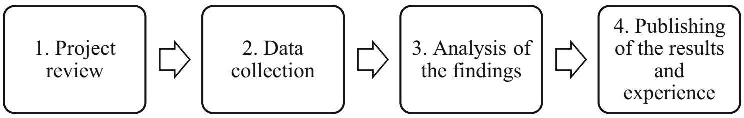

Fig. 1 Steps of the postmortem analysis for small- and medium-size companies (Source: Developed by authors, based on Myllyaho et al. [8])

After initial review of the available information, the authors decided that data collection methods should include in-depth interviews with former Clusterpoint executive-level employees and also an observational method since one of the authors of this chapter was an employee at Clusterpoint company from February 2015 till March 2016.

The main threat considered to the use of the described methods within the particular case study was related to the analysis of the findings and the subjective interpretation by the former employees having no direct intel and insights from the shareholder and board meetings where the strategic issues were discussed.

Such concern was addressed by interviewing a former company board representative who has been also a shareholder at the same time and was present in the board meetings, thus, being able to comment on these aspects.

2.2 What Is a Product/Market Fit?

When talking about product–market fit, it is important to establish a common understanding of this concept. According to Rachleff and Andreesen, product/market fit means being in a good market with a product that can satisfy that market. “You can always feel when product/market fit isn’t happening. The customers aren’t quite getting value out of the product, word of mouth isn’t spreading, usage isn’t growing that fast, press reviews are kind of “blah”, the sales cycle takes too long, and lots of deals never close” [9].

According to Ellis [10], in terms of measuring if a company or startup has achieved product/market fit, it is suggested to ask its early customers the most important question for determining how well its product is resonating with users: “How would you feel if you could no longer use [product]?” given that possible answers are: Very disappointed, Somewhat disappointed, Not disappointed (it is not really that useful), N/A—I no longer use [product]). If most of your respondents are saying that they would only be “somewhat disappointed” without your product, they are really telling you that it is only a “nice to have.” If the company finds that over 40% of users are saying that they would be “very disappointed” without the respective product, there is a great chance one can build a successful business on this “must have” product and product–market fit is likely achieved [10].

Therefore, it can also be concluded that a product/market fit is a mandatory prerequisite for successful scaling of any business model. Otherwise, there is a little or no chance for achievement of sustainable or any growth in the long term, and its “nice to have” category solution without much of a competitive advantage.

2.3 Premature Scaling of a Startup

According to Blank, one of the most common ways due to which startups die is "premature scaling"—a business is scaling prematurely if it is spending significant amounts of money on growth before it has discovered and developed product/market fit [11]. Blank describes why premature scaling can happen: "Ironically, one of the greatest risks is high pressure expectations to make these first pass forecasts that subvert an honest Customer Development process. The temptation is to transform the vision of a large market into a solid corporate revenue forecast before Customer Development even begins" [11–13].

Therefore, possibility of premature scaling should be examined also in case of Clusterpoint development, taking into account the symptoms that describe premature scaling—the fact that 93% of startups that scale prematurely never break the $100,000 revenue per month threshold, and the team size of startups that scale prematurely is three times bigger than the consistent startups at the same stage [11]. According to publicly available financial data [5], Clusterpoint did not and was not even close to breaking this threshold within a time period from January 2015 till November 2017.

3 The Results of the Postmortem Analysis and the Questions Still to Be Answered

The journey of Clusterpoint from early 2015 till the end of 2017 can be compared to the path of a rollercoaster—from promising highs to upsetting lows that ended with an unfortunate crash. The following sections highlight the positive and the negative aspects of the rapid growth, indicating the things that were executed well and contributed to the growth positively in terms of technology development and marketing efforts for the initially selected marketing strategy, and the main negative aspects that faced company before facing the inevitable crush related to unwillingness to implement the necessary pivot and change the business model to align with the market expectations.

3.1 The Positive Aspects: Things that Were Done Well

Rapid Technology Development and Launch of Clusterpoint 4.0 in 2015 According to both, a previous CEO of Clusterpoint (acting in 2015) and a long-time CTO of Clusterpoint, the technical team managed to achieve a considerable technological advancement in terms of Clusterpoint software development, which resulted in the launch of Clusterpoint 4.0 version in October 2015 and the consequent, considerable improvement of the database technology algorithm, within the remaining year. This was fostered by the growth of Clusterpoint team from 10 to 24 people until the end of 2015, mainly related to the expansion of the technical team (software developers and system architects). The management team, including CEO, consisted of 8 people, while technical team, including CTO, comprised remaining 16 people. Company records show [5] that by the end of 2016 there were 28 employees at Clusterpoint company; however, no info is available if this number includes also those who left the company in early 2016 after CEO stepped down. Unfortunately, there is no data available for comparison with similar companies (e.g., MongoDB in its early days) to determine if the headcount of this size at the turnover threshold below US$100,000 per month would have served as an early indicator of premature scaling, according to Blank [11].

Active Promotional Activities and Market Feedback Active offline and online promotional activities of Clusterpoint Cloud in the USA and local Latvian markets served as environment for testing both technology offering (product features) and marketing activities, which essentially meant the testing for the existence of product/market fit. This was implemented by enabling a feedback loop with software developer communities and at the same time facilitating Clusterpoint Cloud trials. Participation at the trade shows along with major NoSQL software vendors like MongoDB, Amazon WS, and IBM Bluemix helped to understand the marketing tactics and sales strategy used by the competitors. Clusterpoint served as one of global supporters (along with Amazon WS, HP, and IBM) for AngelHack hackathon series that were held in more than 70 different cities worldwide (including, USA, Europe, and India) from April 2015 till October 2015 [14]. Clusterpoint also supported several hackathon events and developer focused meetups in other cities and countries (e.g., Database Month in New York, USA; HackingEDU in San Mateo, San Francisco, USA; Garage48 Big Data hackathon in Riga, Latvia; and others). During these events it became clear that NoSQL database marketing is strongly related to relationships with developer communities, and a vendor-like Clusterpoint needs to maintain short proximity to the end user of database technology starting from the software trial until providing extensive documentation resources, online training tools, customer support preferably in the language of end users, and reconfirming credibility of company stability in the long run. The promotional activities helped to generate several thousands of free tier Clusterpoint Cloud user accounts during 2015 and initiate product trials, but none of them was converted to paying customer afterwards, which was a clear sign of a missing product/market fit.

Recognition by Gartner The research and advisory company Gartner included Clusterpoint as a Cool Vendor in platform-as-a-service (PaaS) category in its 2016 report, outlining the vendors that offer new platform opportunities for business and IT, in response to increasing demand for intelligent business operations with cloud levels of scale, agility, and responsiveness [15]. It was followed by Clusterpoint's inclusion in the Market Guide for Database Platform as a Service (dbPaaS) in the same year, where Gartner analysts indicated that the database-platform-as-a-service market continues gaining momentum, driven by maturing products, reports of major resource savings, increased product choice, and a general acceptance of the cloud, suggesting that CIOs and data and analytics leaders should use dbPaaS to address core business requirements [16]. However, it should be noted that Clusterpoint became a paid user for Gartner services and a subscriber for Gartner resources in 2015. This was a tactical step after a senior-level Gartner analyst made some ironic public remarks on his personal Twitter account regarding Clusterpoint's attempt to reach out to him directly for briefing on Clusterpoint technology and NoSQL databases in general. Surely, the recognition by Gartner could not compensate for the lack of product/market fit, but it would serve as a signal to community about seriousness of Clusterpoint's intentions to become a long-term player in the global NoSQL database market.

3.2 The Negative Aspects: Things That Should Be Done Differently

Ignorance of Market Feedback or/and Fear of Pivoting Clusterpoint employees collected customer feedback during trade shows, meetups, hackathons, and tech conferences where Clusterpoint did product demos and interacted with software developers from junior-level developers working for startups to senior IT executives from Fortune 500 companies responsible for developing IT infrastructure at their multi-billion companies. This feedback was communicated regularly over e-mail memos to company board of management. They in turn channeled it further to company shareholders. Already around mid-2015 the customer feedback indicated that Clusterpoint Cloud offering is not appealing to software developers due to the following main reasons: it was based on a closed source code algorithm while majority of competitors, including the leading NoSQL vendor MongoDB, focused on provision of open source code of their on-premises software and aimed to monetize software using general support services (billed by hour) or specific licensing deals for selected enterprise-level customers. Second, the customers were already used to working within the global Cloud infrastructures and platforms-as-a-service provided by Amazon, Microsoft, Google, and IBM. Customers were not interested in locking themselves into a small cloud infrastructure provided by unknown company from Latvia in the Eastern European region, neighboring with Russia. For instance, German customers noted that it is against the law in Germany to keep sensitive data outside their country. Therefore, market and customers demanded open-source database software being hosted and made available as another developer tool within large Cloud ecosystems like within Amazon WS, which was and is used by numerous Fortune 500 level companies and provides possibility of storing data in any of their numerous data centers worldwide. However, Clusterpoint shareholders were clearly against such pivot (a change in strategy without losing the initial vision [17]) and switching from closed source to open-source software because of apparent willingness to maintain revenue from local Latvian market. Next, any attempts to establish relationships and find ways of including Clusterpoint as a developer tool in either of global Cloud platforms were unsuccessful. Any initiative to reach out for Amazon, Microsoft, or IBM representatives during 2015 ended without any response from the global giants. Also, at that time the platforms did not have such elaborate solutions for automated listing of new developer tools and microservices in their marketplaces like they do now for resell to their ecosystem users (e.g., Microsoft Azure Marketplace). The company shareholders decided to continue promoting Clusterpoint Cloud DBaaS service in its initial form, but it did not bring any business results and the only revenue was still coming from enterprise licensing deals of the original on-premises version of Clusterpoint software in the local Latvian market. In June 2016, approximately a half year later, Clusterpoint competitors MongoDB came up with their database-as-a-service offering called Atlas [18], initially available for Amazon WS users, which they have successfully grown over the consecutive one-and-half-a-year into a profitable product category contributing more than 11% of its fourth quarter revenue of US$45 million in 2017 [19].

3.3 *Questions for Further Consideration*

More Time, More Money During interactions with end customers and software developers, it became clear that the vendor support to developer community plays an important role for software developers in decision-making regarding choice of particular developer tool. Also, the involvement of developer community in development and improvement of the open-source software, whose development is basically driven by community activity and engagement. However, it was also clear that Clusterpoint competitors MongoDB did have a considerable advantage in the form of established developer community who prefer working with a supportive vendor and can interact among themselves through online forums and other platforms targeted to open-source software users (e.g., Github). Therefore, it is possible only to guess if Clusterpoint would have succeeded had it more cash and more time for doing another pivot by switching from closed source to open-source format and trying to build its own community to compete with MongoDB, which had pretty similar technical feature list at that moment.

A Startup or an Enterprise? Even though Clusterpoint as a legal company was established in 2006, its co-founders made another large leap of faith with the launch of Clusterpoint Cloud DBaaS offering in 2015, almost 9 years later. On the one hand, the company was already making steady cash flow from its software licensing in the local Latvian market using a tested business model and thus it could be regarded as a traditional enterprise with its management structure. On the other hand, along with Clusterpoint Cloud and subscription-based billing launch the company actually switched to the startup mode, which, according to Blank [20], is a characteristic of an organization formed to search for a repeatable and scalable business model. It is possible that the company management and shareholders attributed former success with on-premises commercialization in the local market to the expectations regarding Clusterpoint Cloud success, but without careful testing of the product/market fit before pouring a substantial amount of venture money into promotion of the DBaaS service. Also, the decision of company shareholders to hire the new CEO in May 2016 with a corporate, not startup background, raises a question about considerations of shareholders and if they did realize that they are betting on "overnight success" in a tough software market with just marginally better product, instead of utilizing Lean Startup approach to make the necessary pivots and arrive to a globally sustainable product/market fit [21].

4 Conclusions

4.1 Implications for Researchers

This case study confirmed the thesis by Rachleff and Andreesen, who stated that the only thing that matters for a startup (which Clusterpoint essentially was at the moment of Clusterpoint Cloud launch in 2015 and during subsequent years) is getting to product/market fit—being in a good market with a product that can satisfy that market [9].

The outcome of Clusterpoint activities showed that the product/market fit was not taking place—the customers were not getting value out of the Clusterpoint Cloud product as only free tier accounts were used; word of mouth was not spreading as only paid PR created some buzz; usage was not growing organically as only Google ads generated free tier sign-ups; company got its product press reviews mainly with the help of its external PR partners; there were no deals closing apart from the ongoing deals for the standard software version being licensed using the traditional licensing model for enterprise customers in Latvia.

Consequently, a prompt change of company direction (a pivot) was necessary instead of trying to push ahead with initially selected strategy and business model. This is also aligned with findings that software startups frequently find that their initial product ideas do not pan out commercially and they must be prepared to change direction in one or more ways implying implementation of a new pivot during early stage of a new product launch to the market [22, 23].

It was concluded that an existing product/market fit in one market (e.g., local Latvian market) for Clusterpoint software was not sufficient grounds to assume existence of product/market fit for slightly different product (Cloud-based DBaaS, not a traditional enterprise software) in another market (e.g., USA) [12].

Clusterpoint company collapse is partially explained by and confirms Blank's theory of "premature scaling." Blank suggested that a business is "scaling prematurely" if it is spending significant amounts of money on growth before it has discovered and developed product/market fit [11, 12]. Clusterpoint case study confirmed that one of important reasons for premature scaling was related to the temptation to transform the vision of a large market into a solid corporate revenue forecast and spending huge promotional budget for facilitating customer base growth before an actual Customer Development took place. Therefore, the decision of Clusterpoint management team to jump into entering of new markets with a new Clusterpoint Cloud product—using a large promotional budget to generate awareness of the Clusterpoint Cloud DBaaS offering before actually validating the existence of product/market fit in the new market—can be regarded as a premature scaling and a partial explanation for failure.

4.2 *Implications for Entrepreneurs*

This case study showed that it is of utmost importance to actively seek for the market feedback and implement brave pivots, if necessary, until achieving product/market fit.

In case of database software commercialization, the case study showed that it was important to consider brave pivots related to switching from closed to open-source format and thus fully change the underlying business model.

It was also concluded that it is important to focus on strategic partnerships with major global Cloud infrastructure providers (Amazon WS, Microsoft Azure, IBM Cloud, and Google Cloud) instead of trying to build own, proprietary Cloud infrastructure, which does not integrate with any of the leading Cloud ecosystems and thus cannot provide any competitive advantages over those database products that focus on their integration with the leading Cloud platforms (e.g., MongoDB) in long term.

The offline marketing activities related to engagement with software developer communities during series of hackathons supported by Clusterpoint were excellent sources for collecting end user feedback regarding the actual user experience on software developer level.

Participation at the trade shows and tech conferences in the USA and various European countries provided an excellent interaction with senior-level IT executives and decision-makers from Fortune 500 companies to understand their mindset and approach for database selection for their company needs from business and technical perspectives.

This was an effective, but comparatively expensive way of collecting valuable customer feedback and getting to realize existing gaps in product/market fit from the user and decision-maker (not always the same person) levels.

Finally, it was concluded that one year is too short period for entering an unknown market and establishing relationships with end customers on Fortune 500 company level for a young startup without prior experience and lack of "warm intros" in foreign markets like the USA. After an intensive year of doing offline and online promotional activities in US market, the Clusterpoint company representatives started to be recognized by USA. IT community members during trade shows and tech events by the end of the first year in the region where it focused its activities (New York and Boston areas). However, it was too early for Clusterpoint to consider US companies as ready for negotiating any business deal whatsoever. Therefore, entrepreneurs should be ready and prepared to spend much longer time doing market research, searching product/market fit, and doing customer development before starting to scale the business model of its startup, while keeping in mind the famous quote by IT industry executives that “Nobody ever got fired for buying IBM” [24].

To conclude, the authors agree with Blank that the inability to achieve product/market fit and get sales numbers growing is not simply a responsibility and a problem of a particular position (e.g., sales or marketing). Rather, it is the challenge for the entire company [25]. Sales and marketing should be scaled only after the founders and a small team have found an MVP with a repeatable sales model AND a product/market fit [26].

4.3 Applicability of Conclusions on Local and Global Levels

The lessons learned from this case study are applicable globally and not only for database market, but also generally for any software startup aiming to enter the global software market. The blindness of the startup investors and willingness to believe their own reasoning instead of carefully listening to the market and its potential customers leads to an inevitable loss by the company to the market dynamics in any market and product category. Moreover, taking into account that technology startups are at the forefront of applying new technologies in practice, similar findings have been identified by both practitioners and academics [27].

The specifics of the database market and Cloud-based solutions (i.e., any software provided as-a-service) assume that it is much easier to get attention of the future customers if your product is aligned or even better, already integrated, within the platforms by the global Cloud infrastructure vendor like Amazon WS and Microsoft WS, which have launched their own and the third-party app and also software developer tool marketplaces for expanding their Cloud ecosystems. Therefore, the lessons to be learned from this Clusterpoint case study are applicable also to any database vendor aiming to launch its Cloud-based database offering for the international customers globally.

Finally, as it was already stated earlier in this section, it is essential to make sure that the product/market fit is actually achieved for the global market and not mistaken what product/market fit in a local market only. Clusterpoint success and business results in the local Latvian market were mistakenly attributed not only to future success in other markets (primarily, the USA, Europe, and India), but also to another (new) and slightly different product offering (on-premises vs. Cloud-based), which eventually led to the entire company business collapse.

Key takeaways

- ✓ Timely pivoting and changing of business model is essential for the survival of any startup searching for product/market fits and furthermore a scalable business model.
- ✓ It is important for database technology vendors to enable integrations with the global cloud ecosystems as soon as possible.
- ✓ There are initiatives contributing to long-term brand awareness but not necessarily generating short-term sales. Do not expect to sell to enterprise companies fast if you are only just an ambitious startup.
- ✓ Do not blame marketing for the faulty product strategy and the indecisiveness of the shareholders to approve a new pivot.
- ✓ Do not hire sales before you have validated an MVP and achieved product/market for it.

Acknowledgments Development of this chapter has been funded by the European Union. PostDoc project Nr. 1.1.1.2/VIAA/1/16/089.

References

1. Butcher, M.: Clusterpoint secures €1m from BaltCap to scale its database for clouds. TechCrunch Homepage. https://techcrunch.com/2012/02/09/clusterpoint-secures-e1m-from-baltcap-to-scale-its-database-for-clouds-2/. Accessed 21 Mar 2019
2. Latvian Private Equity and Venture Capital Journal: http://www.mergers.lv/f/Reports/Latvijas_privata_kapitala_apskats_2016_131216.pdf (2016). Accessed 21 Mar 2019
3. Nelson, B.: Clusterpoint announces new database-as-a-service offering. Tom's Hardware Homepage. https://www.tomshardware.co.uk/clusterpoint-3-database-as-a-service,news-50546.html. Accessed 24 Mar 2019
4. Oliver, A.C.: A new document database emerges from the cloud. Infoworld Homepage. https://www.infoworld.com/article/2940140/clusterpoint-launches-document-database-as-a-service.html. Accessed 24 Mar 2019
5. Crediweb Homepage: https://www.crediweb.lv/CLUSTERPOINT/40003850104/. Accessed 24 Mar 2019
6. The Gazette Homepage: https://www.thegazette.co.uk/notice/2912675. Accessed 11 Apr 2019
7. Company House UK Homepage: https://beta.companieshouse.gov.uk/company/09366103/filing-history?page=2. Accessed 11 Apr 2019
8. Myllyaho, M., Salo, O., Kääriäinen, J., Hyysalo, J., Kosklea, J.: Review of small and large post-mortem analysis methods. In: 2004 International Conference on Software and Systems Engineering and their Applications (ICSSEA) (2004)
9. Andreesen, M.: Marc Andreessen's Homepage. https://pmarchive.com/guide_to_startups_part4.html. Accessed 24 Mar 2019
10. Ellis, S.: Venture Hacks Homepage. https://venturehacks.com/articles/measure-fit. Accessed 2 Apr 2019
11. Blank, S.: It's not how big it is – it's how well it performs: the startup genome compass. https://steveblank.com/2011/08/29/it%E2%80%99s-not-how-big-it-is-%E2%80%93-it%E2%80%99s-how-well-it-performs-the-startup-genome-compass/. Accessed 2 Apr 2019
12. Griffin, T.: Andreessen Horowitz Homepage. https://a16z.com/2017/02/18/12-things-about-product-market-fit/. Accessed 2 Apr 2019

13. Blank, S., Dorf, B.: The Startup Owner's Manual: The Step-by-Step Guide for Building a Great Company, 1st edn. K&S Ranch, Pescadero, CA (2012)
14. AngelHack Homepage: http://blog.angelhack.com/2016/04/21/what-does-an-angelhacker-look-like. Accessed 11 Apr 2019
15. Gartner: Cool vendors in platform as a service. https://www.gartner.com/doc/3313117/cool-vendors-platform-service. Accessed 11 Apr 2019
16. Gartner: Market guide for database platform as a service. https://www.gartner.com/doc/3439539/market-guide-database-platform-service. Accessed 11 Apr 2019
17. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Publishing Group, New York (2011)
18. Lardinois, F.: TechCrunch. MongoDB launches Atlas, its new database-as-a-service offering. https://techcrunch.com/2016/06/28/mongodb-launches-atlas-its-new-database-as-a-service-offering/. Accessed 11 Apr 2019
19. MongoDB Homepage: MongoDB, Inc. Announces fourth quarter and full year fiscal 2018 financial results. https://investors.mongodb.com/news-releases/news-release-details/mongodb-inc-announces-fourth-quarter-and-full-year-fiscal-2018. Accessed 11 Apr 2019
20. Blank, S.: What's a startup? First principles. https://steveblank.com/2010/01/25/whats-a-startup-first-principles/. Accessed 10 Apr 2019
21. Gil, E.: High Growth Handbook, 2nd edn. Stripe Press, San Francisco, CA (2018)
22. Bajwa, S.S., Wang, X., Nguyen-Duc, A., Chanin, R.M., Prikladnicki, R., Pompermaier, A.B., Abrahamsson, P.: Start-ups must be ready to pivot. IEEE Softw. **34**(3), 18–22 (2017)
23. Nguyen-Duc, A., Shah, S.M.A., Abrahamsson, P.: Towards an early stage software startups evolution model. In: 2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 120–127 (2016)
24. IBM Homepage: https://www.ibm.com/ibm/history/ibm100/us/en/icons/personalcomputer/words/. Accessed 20 Apr 2019
25. Blank, S..: https://steveblank.com/2010/02/11/it-must-be-a-marketing-problem/. Accessed 11 Apr 2019
26. Nguyen-Duc, A., Abrahamsson, P.: Minimum viable product or multiple facet product? The role of MVP in software startups, agile processes, in software engineering, and extreme programming. In: Sharp, H., Hall, T. (eds.) Agile Processes, in Software Engineering, and Extreme Programming. XP 2016, Lecture Notes in Business Information Processing, vol. 251, pp. 118–130. Springer, Cham (2016)
27. Berg, V., Birkeland, J., Nguyen-Duc, A., Pappas, I., Jaccheri, L.: Software startup engineering: a systematic mapping study. J. Syst. Softw. **144**, 255–274 (2018)

Triggers of Business Success of IT Startup Owners in Russia

Evgeny Tsaplin and Olga Kosova

Abstract Startups in Russia have a very low survival rate. The main sources of failure are financial and managerial mistakes, as well as the influence of external factors. The purpose of this study was to find out what are the triggers of business success for startups that have passed acceleration programs and survived for 3 years since their launch. We present the results of the case study, which were realized through a qualitative methodology framework. The target population of the study was three owners of startups that participated in acceleration programs and whose startups continued to generate income. The startups researched were located in three cities of Russia: Moscow, St. Petersburg, and Tomsk. We relied on Raheem and Akhuemonkhan's theory of enterprise development as the conceptual framework of our study. Data collection included semi-structured interviews, review and analysis of company documents, reflective journal entries, and direct observation of the business processes in the startups. We analyzed the data using Yin's five-step data analysis process. Data analysis revealed four important themes: the evolution of the entrepreneur, sales strategy, the impact of the acceleration program, and recommendations for accelerators and incubators. Interpretation of the research results can contribute to the survivability of startups in Russia, as well as the development of new successful experiences among entrepreneurs. For those people who are intending to start a business, this research outlines the skills that are necessary to launch a startup.

Keywords Triggers of business success · Business acceleration · Startups · Market entry strategies · Entrepreneurship in emerging economies

E. Tsaplin (✉)
National Research University "Higher School of Economics", Moscow, Russia

Telecom-Project Inc., Dubna, Russia

O. Kosova (✉)
Yaroslavl, Russia

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_19

1 Introduction

The survival rate of startups in Russia is very low; about 70% of them do not survive even 3 years [1]. The low survivability of startups negatively affects the attitude of society toward entrepreneurship, while the social perception of business failure threatens the social identity of the entrepreneur and their chances for successful employment or return to business in the future [2, 3]. Business accelerators and business incubators could assist young entrepreneurs in reaching the level of sustainable development. But in Russia, as of 2018, there was no data on the effectiveness of accelerators. Business incubators and business accelerators in other countries have proven to be effective tools for the development of the innovative ecosystem. However, business incubation and acceleration performance varies in different countries [4].

From results of literature reviews, we assumed that business accelerators can have a positive impact on the survival of startups in Russia. One of the main problems faced by startups in Russia is poor market entry strategies. Owners of startups often lack organizational skills to make their business processes reliable. Our research is aimed at technology startups that passed acceleration programs of the Internet Initiatives Development Fund and stayed on the market for more than 3 years. We used the qualitative multiple case study because it is most suitable for this study. In the course of the study, we wanted to find out what were the triggers of the business success of studied startups that have passed acceleration programs and survived in business beyond 3 years. Furthermore, we wanted to get the answers to the following research questions:

- Which market entry strategies did startups use?
- Which of the owners' personal skills have proven important success factors in the business?
- What sale strategies were efficient?
- How much did accelerators impact on startups' survival?

The basis of our study was semi-structured interviews with entrepreneurs, but we also participated in the review and analysis of company documents, reflective journal entries, and direct observation of the business processes in the startups.

Observations from Russian cases

Startups usually face challenges even before product launching. Early strategic planning is one of the key factors for the successes in later stages. The startup owners need to choose a marketing strategy, hire a team, consider a regional context, and decide whether an acceleration program will be useful to the businesses.

2 Findings from the Related Works

We divided the literature review into three groups to investigate the research problem. We analyzed articles about (a) market entry strategies, (b) factors of business success and failure, and (c) the role of business accelerators in startups' successful market entries.

A market entry strategy is set up before the actual product launch. It guides managers' decisions to promote a product, choose an organizational structure, and get the market niche. Having the entry strategy is extremely important to a startup's survival—an effective one increases the chance of business success [5]. It ensures that new ventures find the proper way to reach their goals. The young entrepreneur should decide what, where, when, why, and how to launch their product. An entry strategy gives a limited number of strategic and tactical alternatives in future business operations, so it is important to make the right choice before it begins [6].

From the review of previous studies, we selected three of the most established theories which may guide startups owners to formulate their market entry strategies:

(a) A traditional theory of industrial organization
(b) A resource-based view
(c) A network theory

The first element is presented by Porter's "five forces" theory; he focuses on the market structure in which firms compete and argues that if a firm is looking for a market entry strategy, it has to first analyze the industry. Porter provides five tools for pre-entry market analysis which are the following: the threat of new entrants, intensity of rivalry among existing competitors, threat of substitute products, bargaining power of buyers, and bargaining power of suppliers. Accordingly, the startup owner should create a protection strategy against these five competitive forces. This involves identifying the strengths and weaknesses of the startup and building defenses of its weakness positions [7]. The second theory is vastly different from Porter's conception. The resource-based view sees a company as a conglomeration of valuable, scarce, and firm-specific resources. The approach focuses on the importance of internal resources to predict the firm's successful market entry [8]. It should be noted that "resources" are not only physical but intangible assets like intellectual property, human capital, technology, and brand [9]. So, according to the resource-based view, deciding on market entry strategy should include the next three steps: identifying unique resources of the firm, finding markets where those resources can earn the highest profit, and deciding on the most effective way to sell a product [6]. The third network theory attempts to combine the two previous approaches. The theory describes how a company's external relationships define, refill, and redistribute a firm's resources, shape its strategies, and, through this, impact its productivity. According to this network theory, business relationships are usually viewed as being comprised of three layers: activity links, resource ties, and actor bonds. Welch et al. suggest "ideas" as one more layer for the understanding of network development [10]. The key analysis element here is interaction. So, the firm's network is not static, but dynamic through connection, collaboration, conflicts, and separation. Later, researchers conclude that combining analytical elements of all three theories is becoming the most relevant tool for evaluating startups' market entry strategies. Specifically, through the complex valuation of the firm resources, external connections, and market opportunities, experts and venture capitalists predict the future economic performance of the startup [11].

We did not limit the literature review to articles only about market entry strategies. We also examined studies about startups' survival and shutting down in the post-entry period, as our empirical study involves the analysis of factors for success after at least 3 years of business operation. Managers and shareholders of businesses vary the definition of "success" [12]. In our study, "sustainability" is the appropriate word to describe the success of a technology startup owner in operating beyond 3 years. Factors of success and failure in business development are two sides of the same coin. We suppose that knowledge of possible business troubles may help startup owners predict failure risks and avoid them. There are some particular methods to precipitate the growth or decline of a firm. Scherger et al. [13] showed that the following financial indicators are the most significant: the remuneration of shareholders, the frequency of contributions, budget control, financial planning, and the search for funding. The evaluation of business processes included the incidence of the use of objects, macroeconomic changes, shifts in the regional economy, productivity, and excess capacity. Several causes had even less impact on possible failures, such as the market reach, advertising and promotion, lack of planning, and external advice [14].

Rauch and Rijsdijk [14], as well as Amankwah-Amoah [15], rely on the theory of human capital. Particularly, they found that the more developed the general and special human capital, the lower the likelihood of failure. Holt [16] examined the impact of organizational innovation in the prevention of failure and found that disruptive, incremental, and system innovations can prevent negative influences and, in turn, prevent a failure.

Regardless, the fact that business failure is not always a bad thing, but a point to learn from, has already become common knowledge. For an entrepreneur to maintain a healthy state of mind after a business failure, many factors are significant. Mandl et al. [3] found that stigmatization from society negatively affected the social activity of the entrepreneur. The higher it was, the more motivated the entrepreneur was to create a new company [17]. This might explain the fact that many entrepreneurs start new businesses even before the actual bankruptcy of their previous venture [12].

Our review of earlier scientific articles on the topic of incubation showed that both business accelerators and business incubators help startups in the initial stages of their development. The main difference is the duration of the programs [18]. Raheem and Akhuemonkhan [14] conducted a thorough research that describes activities of business incubators, acceleration process, and their ecosystems. Moreover, Raheem and Akhuemonkhan [14] considered the key features of business incubators, such as their purpose, types, differences, success factors, and, most importantly, their impact on the acceleration of startups concerning their successful development.

Business incubators implement the following functions: support of economic diversification, marketing of new technologies, entrepreneurship development, job creation, and the development of living standards and providing acceleration programs [19]. Jamil et al. [20] argued that business incubators have a huge impact on the development of the country, as they create jobs, open schools, educate leaders, accelerate startups, and stimulate the economy in general. The efficiency of the accelerator is influenced by the regional context as well. Fehder [21] concluded the higher the networking capabilities and investment activity in the region, the stronger the benefit for startups that participate in acceleration programs in the region.

Another group of researchers, Roseira et al. [22], researched business incubators considering benefits for the entrepreneurs themselves, expectations of entrepreneurs during the choice of incubators, and the level of satisfaction with the outcomes of incubation processes. Startups are attracted to the developed infrastructure of an incubator, such as office equipment and buildings, communications, and business-related and networking services [23]. Each company has its own needs, often unique, which correspond to the stage of its development and the specifics of the product. Incubators that seek to focus on the needs of their residents receive a higher rating [24]. On the other hand, accelerators limit their effectiveness in cases where their startups need mass customization of the product [25]. Moreover, accelerators aim for the quick development of high-growth firms. This quickness is caused by the short-term orientation of accelerators [26]. Accelerators can be less effective for firms in other investment stages [26].

Lai and Lin [27] described how various system indicators of the business incubation process—such as intellectual property, capital, networking, facilities, and equipment—influence the growth of startups. The researchers proposed measurement tools for these system indicators and compared their results with real-life indicators.

We should also emphasize the regional context of the study. The history of business incubation in Russia is over three centuries long. Latov and Latova [28] compared the project with the innovation clusters that existed in Russia in the eighteenth-century mining industry and the twentieth century in the form of "*naukograd*," meaning science cities, after World War II. Nowadays, the Internet Initiatives Development Fund is one of the best performing investment funds and incubation platforms in Russia [29]. The primary objective of the Internet Initiatives Development Fund is to support small and medium-sized enterprises [30]. The fund's activities aimed at supporting startups include three stages: a pre-accelerator, a distance acceleration course, and face-to-face classes. The participants are expected to learn how to draw an investor's attention to their projects [30]. The fund also provides support to the entrepreneurs after they complete the acceleration process. The Internet Initiatives Development Fund is part of a startup ecosystem which, coupled with the development of the Internet, has contributed to the growth of innovation economics in Russia [30]. However, as of 2018, there was no research data about the performance of business accelerators in Russia.

As startups continue to fail at high rates in Russia, research on their market entry strategies and success triggers makes sense. The purpose of the study is to understand what skills owners of newly formed enterprises need to succeed in business and how the business acceleration process contributes to it. Also, the results of our study seem interesting from the perspective of demonstrating the regional context, since the indicators of business incubation differ from country to country [4].

3 Research Method and Design

3.1 Choice of Research Methods

Bryman and Bell [31] stated that research methods, including research design, are fundamental elements that guide the researcher through the whole process. A researcher's decision-making at the initial stage, where they have to choose between qualitative, quantitative, or mixed-methods, will affect the results [31]. For this study, we have chosen the qualitative research method to gain a better understanding of startup market entry strategies. It was not possible to conduct the research using quantitative or mixed methods, as the researcher has limited access to statistical information on nonpublic enterprises [32]. We cannot expect to get a complete picture of market entry strategies, as this requires consideration of larger samples [33, 34]. Nevertheless, we consider that it is useful and important to study cases of specific enterprises as the observations for the practical needs of startup owners and future entrepreneurs.

According to Palinkas et al. [34], a qualitative study can include the following design methods: (a) case study, (b) ethnography, (c) grounded theory, and (d) phenomenology. While all research designs differ from each other, it is up to the researcher to decide which one aligns well with the objectives of the study [34]. We have chosen a case study method as it focuses on a particular situation or a system [34, 35]. To achieve our main research objective, we collected data combining semi-structured interviews, review and analysis of company documents, reflective journal entries, and direct observation of the business processes in the startups. The use of methodological triangulation allowed us to verify data from other distinct points to enhance the trustworthiness of the study results and reduce potential researcher's bias [36]. The qualitative study design consists of a multiple case study to discover the triggers of successful market entry of accelerated technology startups that are in business beyond 3 years.

The target population was startup owners who completed an acceleration program from the Internet Initiatives Development Fund. The startup manager had to be an owner or a shareholder of a company that had successfully graduated from an Internet Initiatives Development Fund acceleration program. There was no income limit for the company, but it had to be in operation and generating revenue. According to the website of the Internet Initiatives Development Fund (IIDF), at the time of research design, there were ten rounds of the acceleration program with a total of 271 participants [37]. We drew our sample of companies that had successfully been operating for over 3 years from this pool of accelerator participants utilizing continuous selection methodology.

3.2 Case Selection

The sample size included three startups from three different Russian cities: Moscow, St. Petersburg, and Tomsk. The sample of our study is not large, but continuous, i.e., we studied all startups which met the selection criteria. Since each case was considered comprehensively, we decided on using more research tools instead of increasing the number of cases. Moreover, our original strategy was to take a more detailed look at each case rather than increasing the number of interviews. That is why, besides interviews, we conducted observation and the examination of company documents and kept the reflective journal. Combining research methods provided us with an opportunity to collect a full database about all selected startups.

The companies we studied operate in the B2B sector. Startups offer their customers innovative products, i.e., services to automate and optimize their business processes to increase sales. Particularly, Startup 1 works on loyalty cards for distributing facilities, Startup 2 offers services to optimize websites for smartphones and add widgets, and Startup 3 manages an online trading platform.

3.3 Data Collection and Analysis

The data collection phase was completed in January–February 2018. Analyses of documents allowed us to prepare for the field research phase: to complete the interview guide and a list of observations. The study of scientific literature helped us to focus on the research framework and choose categories of future analyses. The study of IIDF and participating companies' websites, open data on the activities of startups (archival information, statistical, and tax reports), changes in the employee numbers and managers, and the dynamics of basic economic indicators gave us information about the companies' type and structure and allowed us to collect data on the products that companies offer and to analyze their self-representation in the public sphere. Interviews were recorded, transcribed, and coded. Observation forms were structured by subject. Reflective journal entries allowed us to have a critical approach to data analysis and information structure through a mental map. Moreover, we used Yin's [35] five-step data analysis which involved (a) evaluating, (b) categorizing, (c) organizing, (d) analyzing, and (e) rearranging data to collect observation-based assumptions.

A potential limitation of our study can appear in an inability to transfer the research findings to technology startups in other countries. Additionally, the circumstances for technology startups accelerated by the Internet Initiatives Development Fund may differ from other technology parks, accelerators, and business incubators depending upon the industry in which they specialize. Moreover, work experience of the authors in the field of technology entrepreneurship may have caused bias, allowing them to observe the details that less experienced researchers may miss.

Mixed research methods collect data from

- ✓ review and analysis of company documents,
- ✓ semi-structured interviews,
- ✓ direct observation of the business processes,
- ✓ reflective journal entries.

Delimitations are constraints that are arranged by the researcher to narrow the scope of a study [31]. Our population included a small sample of three participants to represent the acceleration program of the Internet Initiatives Development Fund. Thus, we did not account for technology startup companies that are less than 3 years old. Our study is specific to the Russian innovation ecosystem because of the significant number of small technology startups around the world.

4 Case Study Findings

The purpose of this study was to discover the triggers of accelerated startups' business success and what helped them to survive in the market for 3 years after the launch. We used a semi-structured interview as well as review and analysis of company documents, reflective journal entries, and direct observation of the business processes in the startups. We received lucrative information about the unique and typical experience of Russian technology startups in the first 3 years on the market, as seen in Fig. 1. We identified four significant topics participants most often referred to: (a) evolution of the entrepreneur, (b) sales strategy, (c) acceleration impact, and (d) recommendations for accelerators and incubators.

4.1 Evolution of the Entrepreneur

This section considered (1) the background of the entrepreneurs and (2) the entrepreneurial skills they perceived to gain during the acceleration process of their startups. All three participants graduated from college with a degree in physics or mathematics, with two of them having PhDs in the same field. Moreover, all participants had work experience before they became entrepreneurs. Participants started their new ventures in the same business sectors in which they had previously worked. Such behavior is typical for serial entrepreneurs [38]. In fact, for the

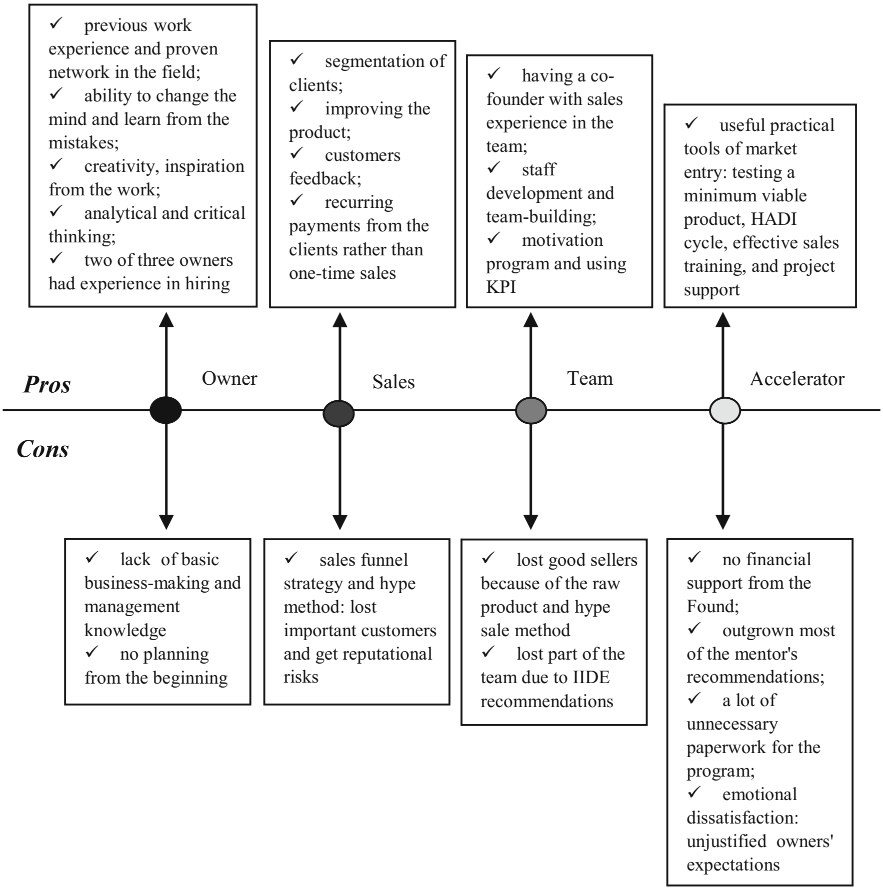

Fig. 1 Business experience of studied startups

participants, moving toward entrepreneurship meant continuing to develop their respective careers within the same field. Thus, they can use their knowledge in the field and proven relationships with technical experts, suppliers, and potential buyers. This corresponds with the networking theory of market entry strategy.

All of the participants mentioned that the challenging educational curriculum had provided a platform for the development of qualities that contributed to successful entrepreneurship, such as analytical and critical thinking, the ability to work intensively with large datasets, and overcoming difficulties. Still, they stated that the knowledge their college education had given them was insufficient for starting a business. They had to develop their managerial skills and ability to work with people to reach success: “*Of course, this is a business advantage to have the people who can critically assess the situation and test hypotheses. But, it is difficult to retrain. I* *had to change myself so much after this deeply technical education to learn how to work with people*" (Participant 1).

Another important fact was that all of the entrepreneurs exhibited active interests in their startups and drew inspiration from their work. Startup owners focused on their projects. Business success means the public acceptance and their self-evaluation, but it takes a lot of effort to achieve their business goals. Participants also mentioned diligence and zeal among the key factors that had helped them. So, Participant 3, speaking about his entrepreneurial experience, remarked: "*the ability to successfully move forward, despite mistakes, is a quality that allows many companies to survive*." All startup owners are convinced that day-to-day self-improvement and broadened horizons are necessary for their business success.

Our research revealed that, as a part of the process of expanding their knowledge of management and in addition to utilizing an accelerator, the startup owners had begun to implement management tools such as customer development and traction. Two of them were unconscious of this implementation, as they did not mention it in the interview. All three startups used customer development concepts [39]. Steve Blank introduced a lean startup customer development methodology, which is an approach to creating new companies, products, and services [39]. The traction concept helps assess how well an entrepreneur's team can implement a project [40].

According to the data analysis, the critical success factor for startups' survival at year one is still the personal qualities of the company owner, namely (a) characteristics, (b) previous experiences, and (c) their abilities to organize their business better than anyone else on the market. The owner's personal performance is the enabler of the initial growth, which allows the startup to launch their products or services. Participant 2 emotionally mentioned: "*A successful, normal businessman, he is, therefore, a successful, normal businessman, because he doesn't want to be a trained little monkey. He decided to build his own life and earn the money*." For success at the next stages of the startups' launch hiring, staff development and marketing strategy become the priority.

4.2 Sales Strategy

The research in this section resulted in three main elements: (1) sales at early stages, (2) hiring sales professionals, and (3) the sales methods used.

In the early stages of a startup's development, it is important to have a cofounder with sales experience in the team—their contacts and reputation facilitate the first sales and help find the first adopters who will believe in the future of the project, even if the product is still coarse-grained. Having an established portfolio of clients from previous workplaces increases confidence in the newly formed company and products [41]. According to company documents and human resource records, during the later startup stages, the cofounder supervised key clients and increased the efficiency of sales.

However, the first customers often get an early version of the product with a good discount. Competent customer segmentation played a key role here. All studied startups focused on a narrow customer segment whose demand could be satisfied with a minimum viable product (MVP) [18]. Later, with incremental enhancement of MVPs, companies reached new customer segments. This practice corresponds to market entry strategies of European software startups [42]. Participant 1 recalled that "*at the beginning the strategy was that, when we do not know anything yet, we offer consulting. All the risks is on the client's side. As soon as we had proven ideas, we offered our clients not only consulting, but also a ready-made solution with software. As a result, nowadays we have a full range of services—from concept development to implementation*." A developed and viable product as a result of the successful first sales allows the hiring of sales professionals in the future—their work will be more effective if they are confident in the product.

The founding teams of all three startups had to hire the first sales professionals when the cofounder responsible for sales could no longer cope with all the tasks. This happened after the first successful sales and receiving feedback on the first products. One of the participants lacked experience in searching for and hiring new team members, while the other two had such experience. Review of company documents showed that after hiring new sellers, the companies' revenues began to grow proportionally. The founders were engaged in maintaining an exciting and motivating atmosphere in the team after more and more people joined it.

It is important to note that startup owners used and refused *the sales funnel strategy*, considering it to be wrong and dead-end. Participant 1 noted that they had used a sales approach based on hype. This approach places a potential client into a state in which they make a purchasing decision based on the emotions and psychological tricks of the seller. However, at some points in the startups' life cycles, company owners completely discarded this method. Renunciation of the hype method of sales occurred because startup owners were emotionally disappointed by this approach and felt guilty. Also, Participant 1 came to an understanding of the impossibility of creating a sustainable business by selling a product that does not have value for a client, as well as the potential reputational risks this strategy entailed. We assume that startup owners learned a lot from the *negative wave* of customers' feedbacks. Perhaps this feedback allowed entrepreneurs to pivot their businesses to sustainable paths by satisfying consumers and improving products.

According to corporate documents, in the early stages of the startup's life cycle, success stories in which a similar solution was put into practice in other companies helped to find the first buyers. The segmentation of clients also played a significant role. All the participants actively used the so-called *hype method* to motivate the first clients, but later completely discarded this method.

Improving the product and staff development are elements of resource-based view market entry. However, customer segmentation links to Porter's "five forces" theory; the *hype method* and cofounders' previous sales experience in the field are discovered elements of the network theory. Thus, the studied companies used mixed market entry strategies throughout the first 3 years. However, the observation of theoretical elements in startups' approaches was neither complex nor well measured.

Startups owners did not reflect on this. In the interview they all accepted that in the beginning startups were led by intuition, not by strategies. They started using deliberate strategies only after receiving some negative experience. Thus, we confirm the results of previous Russian studies about startups' market entry experience, which pointed to a low level of business planning before product launch. We think that it is primarily due to the lack of basic business-making and management knowledge of startup owners. At the same time, many cases of famous software startups show that owners frequently find that their initial product ideas do not pan out commercially. Business success is in being prepared to change direction in one or more ways [43].

4.3 Acceleration Impact

At the stage of creating their startups, the teams were looking for any expert opinions and investors that would help them develop their businesses. All three considered *Skolkovo Innovation Center* as an option but did not use it for various reasons and resorted to the help of the Internet Initiatives Development Fund.

All participants admitted that the acceleration program itself had only little impact on the development of their business, but, in its course, they received new and useful knowledge. Participants noted that they received some common useful practical tools for product development and marketing, such as testing ideas with MVPs, HADI cycle, effective sales training, and project management [44]. However, IIDF declares to attract third-party investments for its startups; during the interview, all startup owners mentioned that they had found funding from other sources in the process or after the end of the acceleration program without using the fund support. Moreover, two out of three participants reported that, although they had gained additional managerial skills during the program, in general, they had already outgrown most of the recommendations offered by mentors. Thus, all startup owners noted that they used the knowledge and experience gained from the acceleration program, but they refer to fund recommendations critically and develop their business according to their visions of the company's prospects. This can be interpreted as a refusal to blindly follow external recommendations in favor of a more balanced and deliberate management decision.

4.4 Recommendations for Accelerators and Incubators

All interviewed participants suggested recommendations for improving the process of acceleration and the work of the Internet Initiatives Development Fund to better develop survival skills in startup owners. The participants suggested that accelerators and incubators should:

- Create a club system and a community of like-minded people: an open platform for a direct interaction between program participants, both residents and graduates, while business and technical experts will allow participants to train their networking skills, find potential business partners and customers, and test and improve their products.
- Offer more transparent working procedures and methods. This will increase the confidence levels of managers, employees, and mentors of the accelerator, as informants regretted the standardized approach to projects, the insufficient experience of mentors in startups' launch, and the subjectivity of recommendations.
- Create several programs depending on the entrepreneur's level of development. Companies which come to IIDF are very different. However, the acceleration program is the same. The recommendations are usually aimed at companies in the early stages; our informants came to the IIDF at the later stage and were not satisfied by the training, because the program did not meet their needs: "*on one hand, you want people like me, not freshmen. And on the other hand, I come to you, and you start teaching me how to walk. I don't need it, I can walk*" (Participant 2).
- Create an ecosystem of suppliers. Entrepreneurs need a contact list of certified and verified suppliers: "*infrastructure is needed, good designers, layout designers, lawyers, programmers, marketers, salespeople. The best in their respect is quality. That will be useful*" (Participant 2). This will save them time and money.
- Provide training and education of entrepreneurial skills. Participants noted the importance of developing startup owners' soft skills, such as responsibility, discipline, self-management, emotional intelligence, critical thinking, effective communication, and teamwork skills. Participants think that mentors should pay more attention to the psychological type of startup managers and select an individual approach for each of them.

5 Conclusions

In our qualitative research, we studied the triggers of business success in technology startups that have undergone acceleration programs and run their business for more than 3 years. Our data collection method was a combination of semi-structured interviews, review and analysis of company documents, reflective journal entries, and direct observation of the business processes in the startups. During the research, we collected data from startup owners who participated in the acceleration program of the Internet Initiatives Development Fund. This study aimed to understand how technology startups can create a strong, competitive, highly effective market entry strategy.

The study revealed the tools that startup owners used for sustainable development during the first years of operation: customer development, problem and solution interviews, and special methods of working with data. Also, we discovered that studied companies used mixed market entry strategies through the first 3 years. However, usage of the elements of the theoretical approaches in companies' strategies was not reflected upon, complex, or well measured. Startup owners accepted that initially their businesses were led by intuition, not by strategic planning. They started adopting deliberate strategies only after negative experiences. We think that it is primarily due to the lack of basic business-making and management knowledge of startup owners.

We found that the researched acceleration program did not have a significant impact on the survival of startups, their marketing strategies, sales strategies, or human resource management processes. However, acceleration programs influenced participants' worldviews, developing the skills of tactical planning. The overall results of the research indicate that the critical factors for startup owners' survival are (a) personal characteristics, (b) previous experience, and (c) the ability to conduct their key business activities. These are the key factors that allow a startup team to successfully launch and reach first paid customers. Marketing tools and human resource strategy are the second layers around these three core factors that speed up the growth in later startup stages.

Keeping in mind the fact that Russia has not yet had empirical research conducted about the effectiveness of business acceleration programs, the results of our work fill up this niche, reveal the pros and cons of startups, and show points of growth for accelerators. Business incubators and accelerators may uncover information on how to adjust and adapt their programs to better develop survival skills among entrepreneurs. The findings of the study could lead to a positive social change among startup owners as well. The results could contribute to increasing the startup survival rate as well as exchanging successful experiences among new entrepreneurs.

For a generalization validity, the results of our research contribute to the study of the regional aspects of launching startups on the market by showing Russian cases. However, the uncertainty of transferring the results of the current study to other countries leads to a recommendation for further research on accelerated startups around the world. Research regarding various acceleration and incubation programs throughout Russia and other countries may uncover valuable information that was not found in this research.

Common shortages in Russian startups

- ✓ lack of strategic planning at early stages
- ✓ lack of fundamental business development and management knowledge,
- ✓ using sales strategies which is inefficient in the long-term,
- ✓ participating in the acceleration program at the late stages of market entry.

References

1. Bezrukova, T., Stepanova, Y., Shanin, I., Durakova, Y.: Sovremennoe sostoyanie i razvitie startapov [Current state and development of startups]. Adv. Curr. Nat. Sci. **1**(1), 95–97 (2015). Available at http://natural-sciences.ru/pdf/2015/1-1/34786.pdf. Accessed 25 Mar 2019
2. Jenkins, A.S., Wiklund, J., Brundin, E.: Individual responses to firm failure: appraisals, grief, and the influence of prior failure experience. J. Bus. Venturing. **29**(1), 17–33 (2014). https://doi.org/10.1016/j.jbusvent.2012.10.006
3. Mandl, C., Kuckertz, A., Allmendinger, M.: Exploring the societal perception of business failure. International Council for Small Business (ICSB). Washington, USA (2015). Available at https://drive.google.com/file/d/0B9pflhVOKMWBWWhGZTJOYXgyUzQ/view. Accessed 25 Mar 2019
4. Bruneel, J., Ratinho, T., Clarysse, B., Groen, A.: The evolution of business incubators: comparing demand and supply of business incubation services across different incubator generations. Technovation. **32**(2), 110–121 (2012). https://doi.org/10.1016/j.technovation.2011.11.003
5. Porter, M.E.: Towards a dynamic theory of strategy. Strat. Manag. J. **12**, 95–117 (1991)
6. Teo, E.: Market entry strategies of wireless startups. Haas School of Business, University of California, Berkeley (2002). Available at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.134.1022&rep=rep1&type=pdf, 22/06/2019. Accessed 25 May 2019
7. Porter, M.E.: The five competitive forces that shape strategy. Harv. Bus. Rev. **86**(1), 25–40 (2008)
8. Teece, D.J., Pisano, G., Shuen, A.: Dynamic capabilities and strategic management. Strat. Manag. J. **18**(7), 509–533 (1997)
9. Peteraf, M.A.: The cornerstones of competitive advantage: a resource-based view. Strat. Manag. J. **14**(3), 179–191 (1993)
10. Welch, C., Wilkinson, I.: Idea logics and network theory in business marketing. J. Bus. Bus. Market. **9**(3), 27–48 (2002)
11. Miloud, T., Aspelund, A., Cabrol, M.: Startup valuation by venture capitalists: an empirical study. Venture Cap. **14**(2–3), 151–174 (2012)
12. Dias, A.R., Teixeira, A.A.C.: The anatomy of business failure. A qualitative account of its implications for future business success. FEP Working Papers, vol. 550, November 2014, pp. 1–22 (2014). Available at https://ideas.repec.org/p/por/fepwps/550.html. Accessed 25 Mar 2019
13. Scherger, V., Vigier, H.P., Glòria Barberà-Mariné, M.: Finding business failure reasons through a fuzzy model of diagnosis. Fuzzy Econ. Rev. **XIX**(1), 45–62 (2014). Available at https://ideas.repec.org/a/fzy/fuzeco/vxixy2014i1p45-62.html. Accessed 25 Mar 2019
14. Raheem, S., Akhuemonkhan, I.: Enterprise development through incubation management. Dev. Countr. Stud. **4**(18), 67–82 (2014). Available at http://www.iiste.org/Journals/index.php/DCS/article/view/15986. Accessed 25 Mar 2019
15. Amankwah-Amoah, J.: A unified framework for incorporating decision making into explanations of business failure. Ind. Manag. Data Syst. **115**, 1341–1357 (2015). https://doi.org/10.1108/IMDS-03-2015-0085
16. Holt, G.D.: Construction business failure: conceptual synthesis of causal agents. Construct. Innovat. **13**(1), 50–76 (2013). https://doi.org/10.1108/14714171311296057
17. Yamakawa, Y., Peng, M.W., Deeds, D.L.: Rising from the ashes: cognitive determinants of venture growth after entrepreneurial failure. Entrepren. Theor. Pract. **39**(2), 209–236 (2015). https://doi.org/10.1111/etap.12047
18. Duc, A.N., Abrahamsson, P.: Minimum viable product or multiple facet product? The role of MVP in software startups. In: International Conference on Agile Software Development. Springer, Cham, pp. 118–130 (2016)
19. AL-Mubaraki, H.M., Busler, M.: The importance of business incubation in developing countries: case study approach. Int. J. Foresight Innovat. Policy. **10**(1), 17 (2015). https://doi.org/10.1504/IJFIP.2015.070054

20. Jamil, F., Ismail, K., Mahmood, N., Khan, N.U., Siddique, M.: Technology incubators and institutional development. Jurnal Teknologi **77**(23) (2015). https://doi.org/10.11113/JT.V77.6702
21. Fehder, D.C.: Essays on the evaluation of entrepreneurship programs. Massachusetts Institute of Technology (2016). Available at https://dspace.mit.edu/handle/1721.1/105082. Accessed 25 Mar 2019
22. Roseira, C., Ramos, C., Maia, F., Roseira, C., Ramos, C., Maia, F.: Understanding incubator value—a network approach to university incubators. Universidade do Porto, Faculdade de Economia do Porto (2014). Available at http://econpapers.repec.org/paper/porfepwps/532.htm. Accessed 25 Mar 2019
23. Marimuthu, M., Lakha, P.: The importance and effectiveness of assistance programs in a business incubator. Probl. Perspect. Manag. **13**(3), 79–86 (2015). Available at https://businessperspectives.org/journals/problems-and-perspectivesin-management/issue-49/the-importance-and-effectiveness-of-assistance-programs-in-a-business-incubator. Accessed 25 Mar 2019
24. Mentink F.J.: Which support services deserve the focus of the incubator? A value creation perspective. University of Twente (2014). Available at http://essay.utwente.nl/64748/. Accessed 25 Mar 2019
25. Battistella, C., De Toni, A.F., Pessot, E.: Open accelerators for start-ups success: a case study. Eur. J. Innovat. Manag. **20**(1), 80–111 (2017). https://doi.org/10.1108/EJIM-10-2015-0113
26. Isabelle, D.: Key factors affecting a technology entrepreneur's choice of incubator or accelerator. Technol. Innovat. Manag. Rev. **3**(2), 16–22 (2013). Available at http://timreview.ca/article/656. Accessed 25 Mar 2019
27. Lai, W.-H., Lin, C.-C.: Constructing business incubation service capabilities for tenants at post-entrepreneurial phase. J. Bus. Res. **68**, 2285–2289 (2015). https://doi.org/10.1016/j.jbusres.2015.06.012
28. Latov Y.V., Latova N.V.: Skolkovo kak innovacionnyj centr: Obshchee i osobennoe (istoriko-komparativistskij podhod) [Skolkovo as an innovation center: general and special (historical comparative approach)]. J. Econ. Regul. **6**(1), 37–45 (2015). https://doi.org/10.17835/2078-5429.2015.6.1.037-045. (in Russian).
29. Russian Venture Investment Market, Results of 2014. Available at http://json.tv/en/ict_telecom_analytics_view/venture-market-in-russia-results-of-1h2014. Accessed 25 Mar 2019
30. Halyavskaya T.V.: Razvitie rossijskogo segmenta seti Internet i formirovanie ehkosistemy startapov kak vzaimostimuliruyushchie faktory innovacionnogo razvitiya ehkonomiki RF [Development of the Russian segment of the Internet and the formation of the startup ecosystem as mutually stimulating factors of innovative development of the Russian economy]. Paper Presented at the 5th International Scientific and Practical Conference, Stavropol, Russia, 16–20 Feb 2016 (2016). Available at https://elibrary.ru/item.asp?id=25720367. Accessed 25 Mar 2019
31. Bryman, A., Bell, E.: Business Research Methods, 4th edn. Oxford University Press, Oxford (2015). 756 p
32. Chen, F., Hope, O.-K., Li, Q., Wang, X.: Financial reporting quality and investment efficiency of private firms in emerging markets. Account. Rev. **86**, 1255–1288 (2011). https://doi.org/10.2308/accr-10040
33. Bannon, W.: Missing data within a quantitative research study: how to assess it, treat it, and why you should care. J. Am. Assoc. Nurse Pract. **27**, 230–232 (2015). https://doi.org/10.1002/2327-6924.12208
34. Palinkas, L.A., Horwitz, S.M., Green, C.A., Wisdom, J.P., Duan, N., Hoagwood, K.: Purposeful sampling for qualitative data collection and analysis in mixed method implementation research. Admin. Policy Ment. Health Ment. Health Serv. Res. **42**, 533–544 (2015). https://doi.org/10.1007/s10488-013-0528-y
35. Yin, R.K.: Case Study Research: Design and Methods, 5th edn. Sage, Thousand Oaks, CA (2015)

36. Heale, R., Forbes, D.: Understanding triangulation in research. Evid. Based Nurs. **16**(4), 98–98 (2013). https://doi.org/10.1136/eb-2013-101494
37. Internet Initiatives Development Fund (IIDP). Портфельные компании ФРИИ (2017). Available at http://www.iidf.ru/fond/projects/. Accessed 11 June 2017 (in Russian)
38. Bauer, A.K.: Learning from business failure—does restarting affect the business model design? Jr. Manag. Sci. **1**(2), 32–60 (2016). https://doi.org/10.5282/JUMS/V1I2PP32-60
39. Haines, J.: Iterating an innovation model: challenges and opportunities in adapting accelerator practices in evolving ecosystems. In: Ethnographic Praxis in Industry Conference Proceedings. EPIC, New York, pp. 282–295 (2014). https://doi.org/10.1111/1559-8918.01033
40. Gonzalez-Uribe, J., Leatherbee, M.: The effects of business accelerators on venture performance: evidence from start-up Chile. SSRN Electron. J. (2017). https://doi.org/10.1093/rfs/hhx103/4104437
41. Franco, M.: Networking as a marketing tool in small companies: a random and informal approach. J. Bus. Strat. **39**(2), 47–55 (2018). https://doi.org/10.1108/JBS02-2017-0020
42. Nguyen-Duc, A., Shah, S.M.A., Ambrahamsson, P.: Towards an early stage software startups evolution model. In: 2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE, pp. 120–127 (2016)
43. Bajwa, S.S., Wang, X., Duc, A.N., Chanin, R.M., Prikladnicki, R., Pompermaier, L.B., Abrahamsson, P.: Start-ups must be ready to pivot. IEEE Softw. **34**(3), 18–22 (2017)
44. Berg, V., Birkeland, J., Nguyen-Duc, A., Pappas, I.O., Jaccheri, L.: Software startup engineering: a systematic mapping study. J. Syst. Softw. **144**, 255–274 (2018)
45. Cohen, S., Hochberg, Y.V.: Accelerating startups: the seed accelerator phenomenon. SSRN Electron. J. March 2014, pp. 1–16 (2014). https://doi.org/10.2139/ssrn.2418000
46. Rauch, A., Rijsdijk, S.A.: The effects of general and specific human capital on long-term growth and failure of newly founded businesses. Entrepren. Theor. Pract. **37**, 923–941 (2013). https://doi.org/10.1111/j.1540-6520.2011.00487.x

Brazilian Startups and the Current Software Engineering Challenges: The Case of Tecnopuc

Leandro Pompermaier and Rafael Prikladnicki

Abstract Brazil is consolidating itself in the world of software startups, both by the strength of the market and by the innovation ecosystems that help these new companies to start and grow. In this chapter, we present the technical challenges that these software startups encounter. We share our experience at Tecnopuc, one specific STP located in the south of Brazil, with more than 170 organizations, from which 90 are startups. We present a set of technical challenges that relate to the following steps in developing an MVP: requirements engineering, product prototyping, architectural design, and software testing. Based on the analysis of these challenges, we reflect on how innovation ecosystems such as science and technology parks (STP) could help startups on addressing the challenges identified.

Keywords Software startups · Software engineering · Software development · Innovation ecosystem · Science and technology park · Brazil · Tecnopuc

1 Introduction

Introducing a new product or service to the market is a complicated exercise with the unknown outcome from customers, as it requires determination, vision, and resources to push it to the market. Besides, competition with existing products and being known in the market are some obstacles that new companies face. Companies need resources such as money to transform a new idea into a product, and it has been difficult for most organizations to ensure that the scarce resources last longer and are used effectively.

There are a growing number of new companies called startups. Such companies develop innovative solutions, and Cooper and Albert first defined startups in 1977 as new hi-tech enterprises [1]. In their study, they analyzed startups from San Francisco to understand their success factors. The startup was later defined as a human

L. Pompermaier · R. Prikladnicki (✉)
School of Technology, PUCRS, Porto Alegre, Brazil
e-mail: leandro.pompermaier@pucrs.br; rafaelp@pucrs.br

A. Nguyen-Duc et al. (eds.), *Fundamentals of Software Startups*,
https://doi.org/10.1007/978-3-030-35983-6_20

institution designed to deliver a new product or service on extreme uncertainty conditions [2]. In this sense, Eric Ries defines the lean startup methodology as being a methodology for managing companies in environments of considerable uncertainty [2]. A subset of startups that has software-based solutions is defined as software startups or digital startups. Literature called software startups as newly created companies with no operational history and as extremely fast in the development of leading technologies [3].

These software startups are increasingly obsessed with delivering software products in an extremely short time so that the products can be validated directly by the end users. For this reason, the use of lean software development methodology and the experimentation of business models have become popular in software startups, especially in the design of the minimal viable product (MVP) [4].

Several startups also believe that being located in innovation ecosystems are among the success factors of their endeavors. Such environments (e.g., clusters or areas of innovation, science and technology parks, and accelerators) are rich in networking opportunities such as events, lectures, activities in collaboration with universities and on connections with partners and possible customers. This scenario helps in development of new ideas because it increases the opportunities for innovation. When there is a collaboration, an initial idea can quickly become more mature and, perhaps, even pivot on a new business model.

In this chapter, we share our experience with software startups in one of such environments: Tecnopuc, a science and technology park owned by PUCRS, a private nonprofit university located in the south of Brazil. PUCRS is among the top three national private nonprofit universities and Tecnopuc was awarded three times the best STP of Brazil. We also take the opportunity to present an overview of how Brazil is stimulating the development of such environments and software startups in general, identifying challenges and opportunities on this topic, including what we believe is one of the main challenges: how to define a minimum set of Software Engineering (SE) practices that startups should adopt during the development of their software products.

2 Software Startups in Brazil: An Overview

As in other parts of the world, software startups have gained attention in the Brazilian market. A survey conducted in 2019 by the Brazilian Startup Association (Abstartups) with several Brazilian startups presented an X-ray of these nascent companies [5, 6]. This study indicates that 77% of Brazilian startups focus on corporate clients, and just under half (45%) participated in some incubation or acceleration program. Almost 70% of the surveyed startups either have no billing (38%) or the annual billing is less than US$50,000 (29%), revealing that the Brazilian market still lacks tools that help entrepreneurs to provide further traction to their business [7]. The survey also showed that there are more than 65 startup communities in Brazil, divided as follows: Central (7 communities), Northeast (15 communities), North (8 communities), Southeast (14 communities), and South (21 communities).

In addition, Brazil has more than 12,000 startups, and Rio Grande do Sul (the state where Tecnopuc is located) has experienced a growth of more than 490% in the number of startups in the last 5 years, ranked as the third place among all Brazilian states.

This growth is related to the maturity of the existing innovation ecosystems, the presence of science and technology parks that contribute to innovation and cooperation between companies and universities, and the location of large and successful local cases such as the Gerdau Group (industry) and Sicredi (bank), as well as reference multinationals like Dell Inc., HP Inc., Hewlett Packard Enterprise, Santander, among others.

Within this scenario of startups growth, science and technology parks (STPs) are presented as innovation ecosystems that can contribute to the various stages of the development of a new business. The purpose of an STP is a relationship between the scientific and the business community, allowing the integration of specific knowledge and skills in order to provide the following results [8]:

- To develop a culture of innovation and competitiveness of companies and knowledge-intensive institutions associated with the park
- To facilitate technology transfer and entrepreneurial skills between academia and the business sector
- To stimulate the creation and development of technology-based companies through incubators and spin-offs
- To promote the sustainable development of the community and region in which it is inserted.

There are currently more than 90 STP projects in Brazil at different stages (conception, development, operation), from which several are already operating. This scenario shows one of the main characteristics of the country, which is the creation and development of innovative communities with strong characteristics that foster business in each region. For example, the technology park located at the Federal University of Rio de Janeiro—UFRJ (southeast of Brazil) has more than 60 institutions installed, including the research unit of one of the leading Brazilian multinational companies Petrobras. This unit has the most modern simulator for drilling oil wells in the country. In the Northeast, Porto Digital is an STP located in the city of Recife (capital of Pernambuco state) with more than 300 companies and 8000 people. Porto Digital stands out as the park that houses business in the areas of games, multimedia, animation, music, and design. Organizations that decide to be part of such ecosystems usually look for environments that are intensive in technology and leverage scientific and technical research capacity and laboratories to build competitive advantage through innovation.

As mentioned earlier, the southern state of Brazil (Rio Grande do Sul) houses several STPs including Tecnosinos and Tecnopuc that together host more than 250 organizations and 12,000 people. Altogether, these four STPs houses more than

50% of the companies located at STPs in Brazil, including more than 300 startups. They also received the award of the best STP in Brazil promoted by the Brazilian Association of Science Parks and Areas of Innovation—Anprotec (Porto Digital and Tecnopuc were awarded three times, Tecnosinos twice, and UFRJ STP one time). The next section illustrates these rich innovation ecosystems by providing more details about one of awarded Brazilian STPs: Tecnopuc.

3 The Case of Tecnopuc

PUCRS Science and Technology Park (Tecnopuc) is a modern innovation system that was founded in 2003 and houses businesses of various sizes, associations, and university labs, thus facilitating joint technological development. It brings together major companies that act globally through its technological development as well as startups. The success of Tecnopuc is based on solid factors such as the political, social, and economic status of the city of Porto Alegre, the capital of Rio Grande do Sul State. There are 1.5 million inhabitants in Porto Alegre, though in the metropolitan region this figure rises to 4 million. The city holds a privileged geographic location in MERCOSUL. The metropolitan area offers great potential for business and good infrastructure for scientific and technology development, housing three major universities (PUCRS, UFRGS, and UNISINOS) top ranked by Brazilian official agencies such as the Coordination of Improvement of Higher Education Personnel (CAPES). PUCRS currently has a total of 38,000 students and has graduated over 165,000 students up till now.

The presence of a science and technology park at PUCRS is the result of the university efforts to increase the number of Research, Development and Innovation (RD&I) cooperative projects with industry partners, which stimulates innovation and entrepreneurship. This offers an environment where PUCRS researchers can work with applied research and stimulates the technology transfer to the market through patents or startups development. It is an inherent goal of Tecnopuc to insert PUCRS in the technological, economic, and social development process of the region and the country. The initiative also aims:

- To attract RD&I companies to work with the university
- To promote startup and spin-off technology-based companies
- To attract technological and developmental research projects
- To encourage business–government–university innovation and interaction
- To generate positive synergy between academia and business
- To act in a coordinated manner with the local, state, and federal governments.

Tecnopuc is spread in three campus sites in a total area of 27 ha with the main campus locating in PUCRS Central Campus. Tecnopuc is a multi-industry science and technology park focusing on four main areas: (1) information technology and communication, (2) energy and the environment; (3) life sciences and biotechnology; and (4) creative industry. Tecnopuc is home to more than 170 organizations and 7100 jobs. One of the main focuses of Tecnopuc is startups, and in its strategic plan updated in 2018 it has defined the intention of generating 1000 startups in 10 years. Therefore, an area dedicated to the development of innovative enterprises was created and called Tecnopuc Startups.

3.1 Tecnopuc Startups

Since its foundation in 2003, Tecnopuc has helped in the development of almost 400 startups (151 preincubated, 158 incubated, 13 accelerated, and 84 graduated companies). Tecnopuc currently hosts approximately 90 startups at different maturity levels. One of its main successes is a company called GetNet, one of the largest in the development and management of electronic payment solutions in the country, which was sold to the Santander group for over R$2 billion (approximately US$500 MM).

A part of these numbers of startups is based on a constant focus on integrating entrepreneurial education into the university curriculum. As an example, PUCRS has established an Interdisciplinary Entrepreneurship Laboratory (IDEAR) which works on entrepreneurship as a competence that involves mobilization of knowledge, skills, and attitudes, the exercise of creativity, critical thinking, and the exercise of autonomy. IDEAR helps on developing entrepreneurship skills into the university curriculum, allowing the emergence of young entrepreneurs and stimulating them to advance their ideas. An example is a program called Track Startups PUCRS, executed in partnership between IDEAR and Tecnopuc to transform undergraduate course projects ideas into entrepreneurship opportunities and startup creation.

In the context of Tecnopuc, startups are companies based on innovative business models, services or products with economic, social, or environmental impact. These companies are not necessarily based on the university's intellectual property. Tecnopuc Startups provides the support and conditions necessary for innovative businesses to enter the market sustainably and competitively. Its primary goals are to support the startup development process, provide value-added solutions for startups, empower and develop entrepreneurial skills and attitudes, prospect and capture new entrepreneurs, potential new ventures, promoting internal and external connections to the university, and also strategic partners for startups, and stimulate the entrepreneurial capacity of the PUCRS community. For example, if a PUCRS student has an idea and want to go further and turn it into startups, Tecnopuc offers two main programs: Startup Garage (a pre-startup program where one has the chance to participate in a business modeling program) and the Startup Development Program (a program composed of five distinct phases: Start, Build, Establish, Scale, and Consolidate).

Startup Garage The Startup Garage program prepares successful future entrepreneurs by assisting them in transforming an initial idea into a business with great potential. Various tools are presented to entrepreneurial teams by a group of mentors in a program that includes ideation, validation, and prototyping. The focus is on the development and instrumentation of new entrepreneurs from modeling their business. It is a 3-month immersion at Tecnopuc entrepreneurial and innovation ecosystem, which allows the entrepreneurs to experience the reality of generating new business, work on their entrepreneurial profiles, relationships with partners and mentors, and self-knowledge, all the basis for creating an entrepreneurial attitude. To this end the methodologies of lean startup, customer development, and business model canvas are combined and used as reference. The Startup Garage culminates in a pitch day—an event that brings together mentors, potential investors, and members of the PUCRS entrepreneurship and innovation ecosystem, who assist and evaluate the business of program participants. For those who want to continue in their endeavor, they have the opportunity to apply for the startup development program.

Startup Development Program For up to 30 months this program was structured to make nascent businesses more sustainable as well as generate higher value for society. Five distinct phases lasting six months were created to meet the characteristics of the development of startups and their levels of maturity (Fig. 1).

The start phase is the first moment of an entrepreneur or company in the Tecnopuc ecosystem. It is believed that the aura of the entrepreneur and his/her relationship must be assisted so that the profile of the partners can be monitored, the business model validated, and the company formally constituted. The build phase intends to enable the company to achieve its fit in the market. If this is not possible, the company can swing your business, changing its strategy and conducting further testing and validation. The establish phase allows the validation of the business model in which the entrepreneur becomes sure of the position that the company will assume on the market. The scale phase is aimed at helping the company grow so that its strategy is geared toward scale gains. And the consolidated phase expresses expansion with higher revenues, more employees, a higher number of products and services, and business internationalization strategies.

In addition to the Startup Garage and the Startup Development programs, Tecnopuc also has additional initiatives that help startups in their process, such as:

- The Startup CorpLab program, which seeks to identify and establish a partnership between Tecnopuc and large companies or corporations. This program allows startups and large companies to cocreate solutions to challenges identified by the latter.
- The Internationalization program, which takes the advantages of Tecnopuc international network to identify opportunities for startup international growth. The park has bilateral agreements with several STPs around the world in countries such as Germany, Canada, China, Colombia, Ecuador, the USA, Italy, Portugal, and Russia.
- The Acceleration of synergies program, which focuses on the connection and interaction among all companies located at Tecnopuc. The program aims to identify the needs and potential of each organization, from startups to consolidated operations, and provide opportunities for connections and relationships among all companies.

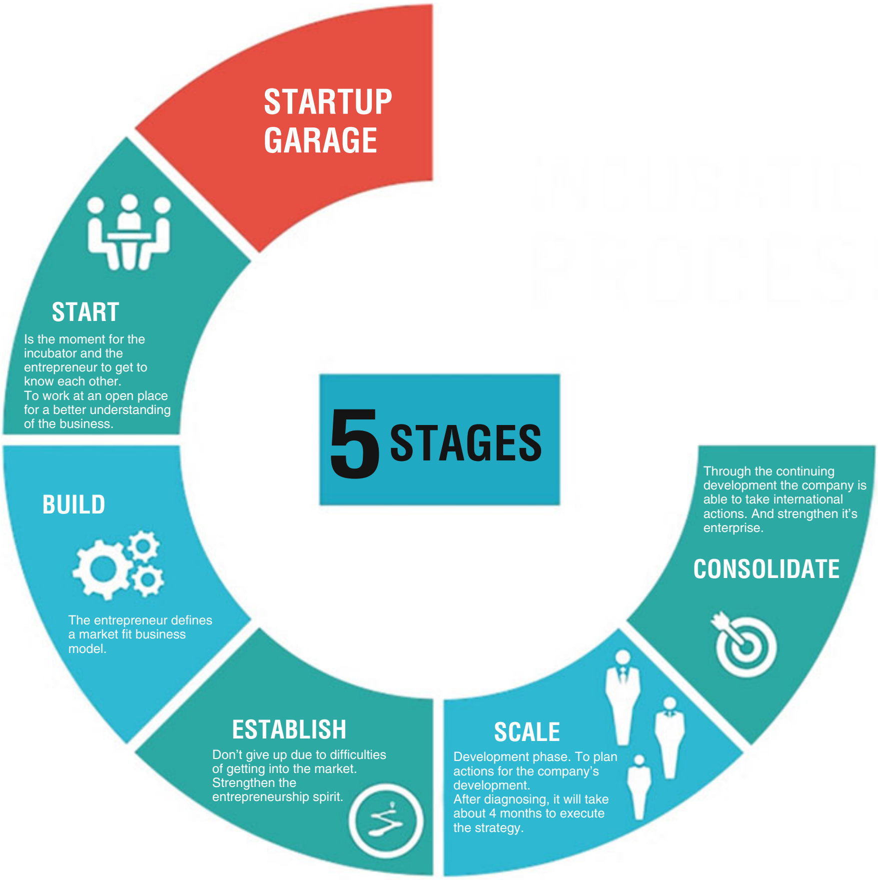

Fig. 1 Phases of Tecnopuc Startup development program

4 Software Engineering Challenges for Tecnopuc Startups

Approximately 80% of all startups located at Tecnopuc are software based, that is, software is in the heart of the company business (software-ubiquity). On analyzing the software startups created and being developed at the park, one can identify several challenges that such companies face. The development of the software product and all activities involved is probably the greatest challenge that software startup entrepreneurs face. This challenge relates primarily to the software development process itself.

For this reason, during the startup development program in 2019, we interviewed a set of 63 entrepreneurs from software startups located at Tecnopuc. We have asked them about their practices on developing their MVP, including requirements, prototyping, architecture design, and testing. We also asked questions related to

Table 1 Startups interviewed and their industry domain

Industry domain	# startups interviewed	%
FinTech	7	11.11
HealthTech	6	9.52
AgriTech	5	7.94
Services	5	7.94
Information technology (IT)	5	7.94
HRTech	4	6.35
Creative industry	4	6.35
Sales and marketing	4	6.35
Management	4	6.35
Logistc	4	6.35
EdTech	4	6.35
PropTech (real state)	3	4.76
Sports	3	4.76
Entertainment	2	3.17
Energy	2	3.17
FoodTech	1	1.59

demographic information of startups (number of jobs created, for example). Our goal was to capture their perception of the use of software engineering practices during their product development, focusing on the early stages of product and company. Within the sample used in these interviews, 48% of the startups use the Business-To-Business (B2B) business model, that is, one business makes a commercial transaction with another. This typically occurs when a business needs the services of another for operational reasons or a business resells goods and services produced by others. Moreover 33% of the startups use the Business-To-Business-To-Consumer (B2B2C) business model, that is, where usually online businesses and portals reach new markets and customers by partnering with consumer-oriented product and service businesses.

In addition, 88% are startups that already have users/clients in their platforms, and 12% are in the MVP's validation phase. Table 1 presents the classification of all startups interviewed ordered by their industry domain. Forty-five percent of the startups interviewed are from five main industry domains—finance (FinTech), health (HealthTech), agriculture (AgriTech), services, and IT (Information Technology).

Based on these interviews, we have mapped the main challenges that startups have encountered and still find related to the use of software engineering practices in the development of their MVPs. The main challenges encountered are related to the following phases of the software development methodology: requirements engineering, product prototyping, architectural design, and testing.

Requirements According to Zowghi [9], requirements elicitation is the process of finding, discovering, acquiring, and elaborating requirements for computer-based systems. However, in a traditional approach the requirements elicitation seeks the characteristics of the client's needs that the solution must contemplate, being critical, and may jeopardize all subsequent steps in case of failure in its activities (increase in cost and development time, cancelation of the project, etc.) [10].

In a scenario where software startups work on defining the requirements of their solutions, problems with defining requirements grow exponentially due to the fact that:

- The customer is not fully defined.
- The problem was not specified and/or defined.

These two essential points for defining software requirements are worked out as hypotheses within the context of software startups and thus should be validated and streamlined when necessary.

In this sense, the requirements discovery should be composed of a combination of the first steps of customer development [11] that consist of the phases of Customer Discovery and Customer Validation. In our software startup program, these steps are represented by hypothesis validation. These activities may be supported by different discovery approaches, such as Design Thinking or Design Sprint. Especially in the Knowing and Creating phases of Tecnopuc's startup development program, a mix of monitoring and mentoring in each interaction with startups helps entrepreneurs address this hypothesis validation.

Startups may use several tools/methods to help them elicit requirements. Examples are interview (talking to potential customers and understand their problems), social media page (generating content in order to drive traffic), blogging (similar to social media but as a blog), and a landing page (a webpage the explains the value proposition and tries to collect data).

The decision on which tool/method to use is done by each development team. According to Melegati [12], software startups do not follow a specific activity when it comes to requirements engineering and is usually influenced by their founders, software development managers, developers, business models, market, and the ecosystem. In addition, the business model is a deciding factor in the choice of practices for requirements engineering [12].

Therefore, requirements engineering that includes elicitation, analysis, documentation, and revision must be adapted according to the maturity of the startup and its team and also aligned with the practices used in the process of Customer Development. Moreover, by giving and receiving feedback software startups can increase the chances of understanding who the customers are, what problems they have, and solutions that may solve their problems.

The survey reported that 63% of startups use Agile practices to manage their requirements, such as documenting requirements using User Stories. They also use visual tools to keep track of the requirements. However, most startups do not set standards in the use of these practices, performing much of the requirements management verbally and informally.

> Main RE concerns with Tecnopuc startups are (1) lack of process formality and (2) lack of documentation.

Product Prototyping All aspects of software development from the earliest stages to system maintenance involve specifying, developing, managing, and evolving software systems. Software Engineering is designed to solve software system problems in order to support development using processes, methods, techniques, and tools [13]. The reality is no different in startups. A common practice we found in startups is the development of product prototyping through short cycles of MVP development.

A prototype is an early sample, model, or release of a product built to test a concept or process. It is a term used in a variety of contexts, including semantics, design, electronics, and software programming. System analysts and users generally use a prototype to evaluate a new design to enhance precision. Prototyping serves to provide specifications for a real, working system rather than a theoretical one. In other words, prototypes help on simulating working software much earlier in the cycle. This can be done in small cycles of MVP development and allows the startup to have the minimum necessary to achieve its goals of quickly seeking market feedback [14].

As an example, once the backlog is managed and priorities are defined, the MPV cycle is planned, including the product prototype that will be delivered for feedback. Moreover, as mentioned by Nguyen-Duc and Abrahamsson [14, 15], parts of what was developed in the MVP can be used in the future for other purposes (communicate with investor, for instance).

Another benefit of developing a prototype through an MVP cycle is the fact that the team becomes more engaged about the needs and the necessary feedback in terms of what should be built and what can be built. In other words, product prototyping reinforces the importance of individuals and interactions over process and tools, which is part of the principles written in the Agile manifesto [16]. In the context of software startups, the adoption of Agile methodologies has been a common practice because they present characteristics that can easily be customized according to the team profile. Agile methodologies are a set of values, principles, and practices based on an iterative and incremental process [16]. The four key characteristics of all Agile methodologies are iterative and evolutionary development, flexible response to change, promoting communication, and the delivery of added value to the customer in shorter iterations [17].

A study conducted by Yau [18] found that different Agile practices are used in different software startups. Speed-related practices are used to a greater extent compared to quality-related practices [18]. The communication practices represented by daily standup meetings are limited adopted because they involve very small teams working together [18]. The survey reported that most entrepreneurs are building their first venture and thus have no experience in running a business. We analyzed the adoption of software development practices, which were used to facilitate and optimize all work done during the MVP's development.

Main challenges during the product prototyping are (1) lack of knowledge on MVP development process and (2) lack of awareness and usages of MVP tools.

Table 2 Challenges with software structure and architecture

Challenges	Description
Coding language and associated IDE	The choice of programming language may affect future extensions of the software and integrations with existing solutions in the market
Database	Need to choose a database that has an active user community, allowing a search for solutions to problems encountered while using MVP
Web host	Web hosting is critical for the solutions developed by software startups. It needs to be an affordable and secure framework
Code deployment tool	Always making software changes and the associated implementations that come with it need to be controlled to increase team productivity and quality
Security approach	Usually indicated as an important item by the startup team but ignored during development (using the security solutions offered by the web host)

Architectural Design Software architecture involves defining a structured solution that meets all technical and operational requirements while optimizing common quality attributes such as performance, security, and manageability. It involves a series of decisions based on many factors, and each of these decisions can have a considerable impact on the quality, performance, ease of maintenance, and the overall success of the software. Table 2 presents the challenges with software structure and architecture faced by the surveyed startups.

Software Testing At each cycle, all development carried out by software startups should be based on responses received by the market, comparing them with the assumptions made in the early stages. Startups should thus be able to design and carry out previous activities quickly and effectively releasing MVP as soon as possible. After that, an analysis of what is happening and how the market is realizing the MVP should be controlled by the startup in order to generate the required learning at this stage.

According to Moogk [19], the key principles of the lean startup include omnipresence of entrepreneurs, uniqueness of the management style of startups, and learning from product testing against relevant metrics. Some measures that entrepreneurs can use are:

- Customer interviews
- Usability testing
- Split testing
- Usage monitoring
- Funnel analysis

Analyzing these data and customer- and market-related learning will help entrepreneurs make important decisions that will lead to a new cycle: increment of

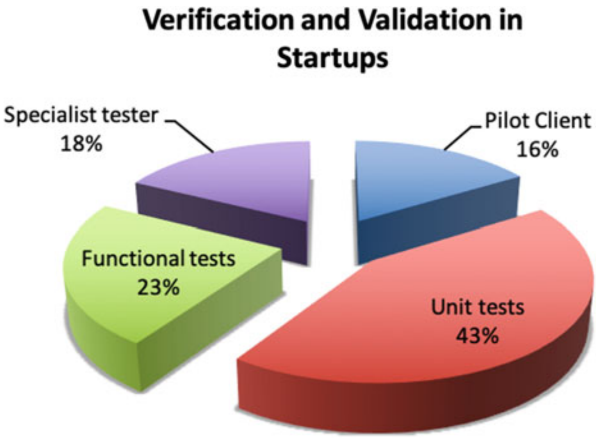

Fig. 2 Portion of software testing in startups—TecnoPuc case

the MVP with new features or flows, creating a new MVP by changing the customer or market focus, or making changes to the implemented functionalities.

> Relevant architectural challenges are (1) lack of knowledge on the deployment environment, (2) lack of built-in quality attributes, and (3) limited configurations.

In the early stage, Tecnopuc Startups performed testing in several ways:

- Using a pilot client (adopted by 16% of total startups): It is considered here the system test performed by a group of end users, prior to deployment, to provide feedback to the product development team [20].
- Unit tests (adopted by 43% of total startups): a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use [20, 21].
- Functional ad hoc tests (adopted by 23% of total startups): software testing performed without planning and documentation [22].
- Specialist tester (adopted by 18% of total startups): software test performed by a business specialist associated with the startup.

The portions of each type of testing are summarized in Fig. 2. As much as startups perform validation and verification activities with their MVPs, only 66% of the investigated cases use some sort of software testing techniques. If we exclude the development activities of unit testing, carried out by the developers, the actual testing on a system level is held by 57% of the respondents, but without any planning or technical guidance.

> Testing concerns are (1) lack of focus on quality assurance and (2) insufficient investment on testing.

5 Conclusions

5.1 *Lessons Learnt from Brazilian Startups*

There are several lessons that can be learned from this study on software startups, both from the perspective of software engineering challenges and the existence of innovation ecosystems that can assist startups in building their solutions. Here we list all lessons we learned.

Lesson 1: Innovation ecosystems such as Tecnopuc are facilitators of the startup development process. Innovation ecosystems help new ventures far more than physical space. It is a catalyst for connections between key players that drive innovation, including governments, society, and companies. These connections provide the new entrepreneurs with opportunities that would be hard to achieve without associating with an ecosystem.

Lesson 2: The MVP's development process is the main challenge for nascent ventures. On interacting with startups, it became clear that the lack of a guide setting minimum standards in the process of developing MVPs leads to technical debt that can be costly in the startup's growth stages. In an environment such as Tecnopuc, professionals from other companies and professors and researchers from the university can be connected to startups to assist them in using best software development practices.

Lesson 3: Identifying early adopters is a crucial contribution of innovation ecosystems. Tecnopuc currently has around 7000 professionals working in various companies, organizations, and startups. This audience can be activated by startups, who are always looking for validations with customers/users. Besides, the university where Tecnopuc is located (PUCRS) has around 30,000 people, including students, professors, researchers, and the community as a whole, which also helps in finding these first clients.

Lesson 4: Defining software infrastructure for their solutions is dynamic and not necessarily based on preestablished industry standards. A startup often struggles to find the right environment for building and validating its MVP. Both Tecnopuc and PUCRS have laboratories that can be used for such validations, such as maker's space, high performance computing lab, among others.

5.2 *Final Words*

Software startups are at the heart of the new economy of this century, which is based on high-impact entrepreneurship, transforming intensive knowledge into exponential innovation. Many startups develop complex software and take advantage of being connected with innovation ecosystems.

In this chapter, we presented our view of this movement by sharing some information about the Brazilian scenario for software startups. We presented the case of Tecnopuc, a science and technology park that hosts several startups, several of them being software startups. We also discussed technical challenges that startups may face from a Software Engineering point of view. We identified a total of nine challenges relating to four different areas of software development.

We believe that it is necessary to continuously find ways to overcome the technical challenges identified. Since Tecnopuc is located at PUCRS University campus, startups can easily connect to SE professors and researchers to better understand and improve their software development process and work on useful documentation. Moreover, relationship with other startups and multinational companies can also contribute to overcoming some of the technical challenges, as companies may exchange experiences and learn from each other on which development environment to use and also exchange quality assurance practices. Defining an MVP development process can also contribute to address challenges related to the lack of knowledge in this area.

In summary, innovation ecosystems such as Tecnopuc can play an essential role in helping startups in several aspects of their process. Otherwise, startups may find an effective way to prototype their solutions and evaluate their market fit, but technical challenges will not only introduce technical debts but also make it very hard for them to scale up.

Acknowledgements This work is partially funded by FAPERGS (17/2551-0001/205-4) and CNPq.

References

1. Cooper, A.C., Bruno, A.V.: Success among high-technology firms. Bus. Horiz. **20**(2), 16–22 (1977)
2. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Books, New York (2011)
3. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: a systematic mapping study. Inf. Softw. Technol. **56**(10), 1200–1218 (2014)
4. Nilsson, H., Linus, P.: How to manage technical debt in a lean startup. Thesis, University of Gothenburg (2013)
5. https://abstartups.com.br. (in Portuguese)
6. https://startupbase.com.br. (in Portuguese)
7. http://ecossistemasdestartups.com.br/ (in Portuguese)
8. Audy, J., Knebel, P.: Tecnopuc – people, creativity and innovation. Edipucrs. http://www.pucrs.br/tecnopuc/livrotecnopuc/en/#home (2016)
9. Zowghi, D., Coulin, C.: Requirements elicitation: a survey of techniques, approaches, and tools. In: Aurum, A., Wohlin, C. (eds.) Engineering and Managing Software Requirements, pp. 19–46. Springer, Heidelberg (2005)
10. Christel, M.G., Kang, K.C.: Issues in Requirements Elicitation (No. CMU/SEI-92-TR-12). Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA (1992)

11. Blank, S., Dorf, B.: The Startup owner's Manual: The Step-by-Step Guide for Building a Great Company. BookBaby, Cork (2012)
12. Melegati, J., Goldman, A., Kon, F., Wang, X.: A model of requirements engineering in software startups. Inf. Softw. Technol. **109**, 92–107 (2019)
13. Sommerville, I.: Software Engineering, 9th edn. Pearson, Boston, MA. ISBN: 10 137035152 (2011)
14. Nguyen-Duc, A., Wang, X., Abrahamsson, P.: What influences the speed of prototyping? An empirical investigation of twenty software startups. In: Baumeister, H., Lichter, H., Riebisch, M. (eds.) Agile Processes in Software Engineering and Extreme Programming, pp. 20–36. Springer, Heidelberg (2017)
15. Duc, A.N., Abrahamsson, P.: Minimum viable product or multiple facet product? The role of MVP in software startups. In: Sharp, H., Hall, T. (eds.) Agile Processes, in Software Engineering, and Extreme Programming, International Conference on Agile Software Development, pp. 118–130. Springer, Cham (2016)
16. Sharma, S., Sarkar, D., Gupta, D.: Agile processes and methodologies: a conceptual study. Int. J. Comp. Sci. Eng. **4**(5), 892 (2012)
17. Maher, P.: Weaving agile software development techniques into a traditional computer science curriculum. In: 2009 Sixth International Conference on Information Technology: New Generations, pp. 1687–1688. IEEE (2009)
18. Yau, A., Murphy, C.: Is a rigorous agile methodology the best development strategy for small scale tech startups? (2013)
19. Moogk, D.R.: Minimum viable product and the importance of experimentation in technology startups. Technol. Innov. Manag. Rev. **2**(3), (2012)
20. Burnstein, I.: Practical Software Testing: A Process-Oriented Approach. Springer Science & Business Media, New York (2006)
21. Myers, G.J., Sandler, C., Badgett, T.: The Art of Software Testing. Wiley, Hoboken, NJ (2011)
22. Chhabra, N.: Introduction to adhoc testing. Int. J. Sci. Technol. Res. **1**(7), 66–67 (2012)