Sami Hyrynsalmi, Arho Suominen, Christopher Jud,
Xiaofeng Wang, Jan Bosch, Jürgen Münch (Eds.)

Proceedings of

International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms (SiBW 2018)

Held in Espoo, Finland, December 3, 2018

Addresses of the editors:

Sami Hyrynsalmi
Tampere University of Technology
P.O. Box 527, 33101
Tampere, Finland
sami.hyrynsalmi@tut.fi

Arho Suominen
VTT Technical Research Centre of Finland
P.O. Box 1000, 02044
VTT, Finland
arho.suominen@vtt.fi

Christopher Jud
University of Stuttgart
Keplerstr. 17, 70174
Stuttgart, Germany
mail@christopher-jud.de

Xiaofeng Wang
Free University of Bozen-Bolzano, Dominikanerplatz 3
– piazza, Domenicani, 3
39100 Bozen-Bolzano, Italy
xiaofeng.wang@unibz.it

Jan Bosch
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
jan@janbosch.com

Jürgen Münch
Reutlingen University,
Herman Hollerith Center
Danziger Straße 6, 71034
Böblingen, Germany
juergen.muench@reutlingen-university.de

Contents

Organizing Committee

Jan Bosch — Chalmers University of Technology, Sweden
Sami Hyrynsalmi — Tampere University of Technology, Finland
Christopher Jud — University of Stuttgart, Germany
Jürgen Münch — Reutlingen University, Germany
Arho Suominen — VTT Technical Research Centre of Finland
Xiaofeng Wang — Free University of Bozen-Bolzano, Italy

Program Committee

Matthias Deschryvere — VTT Technical Research Centre of Finland
Arash Hajikhani — VTT Technical Research Centre of Finland
Ari Helin — University of Turku, Finland
Jukka Huhtamäki — Tampere University of Technology, Finland
Sami Hyrynsalmi — Tampere University of Technology, Finland
Slinger Jansen — Utrecht University, the Netherlands
Tapani Joelsson — University of Turku, Finland
Christopher Jud — University of Stuttgart, Germany
Miika Kumpulainen — Tampere University of Technology, Finland
Markku Kuusisto — Tampere University of Technology, Finland
Jürgen Münch — Reutlingen University, Germany
Nina Rilla — VTT Technical Research Centre of Finland
Rodrigo dos Santos — Universidade Federal do Estado do Rio de Janeiro, Brazil
Marko Seppänen — Tampere University of Technology, Finland
Kari Smolander — Lappeenranta University of Technology, Finland
Krista Sorri — Tampere University of Technology, Finland
Kaisa Still — VTT Technical Research Centre of Finland
Arho Suominen — VTT Technical Research Centre of Finland
Sampo Suonsyrjä — Tampere University of Technology, Finland
Marja Turunen — University of Turku, Finland
George Valença — Universidade Federal Rural de Pernambuco, Brazil
Xiaofeng Wang — Free University of Bozen-Bolzano, Italy
Krzysztof Wnuk — Blekinge Institute of Technology, Sweden
Katariina Yrjönkoski — Tampere University of Technology, Finland

Preface

The very first International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms (SiBW 2018) was held in Espoo (Greater Helsinki), Finland on December 3rd, 2018 – just a day before SLUSH 2018, the world's biggest startup event. Thanks to the collaboration with the organizers of SLUSH, many of the software-intensive business researchers and practitioners took part also in this event.

The international workshop gathered together 35 registered attendees, from Sweden, Germany, Latvia, Finland, Italy and the Netherlands representing both academia as well as industry. The event itself was sponsored by VTT Technical Research Centre of Finland and the workshop was organized by the newly founded Software-intensive Business research community (c.f. [1]) together with Software Startup Research Network (SSRN)[1].

This year's workshop consisted of 19 workshop papers and a seminal keynote. All papers submitted to the workshop were reviewed by at least two members of the program committee and the papers were selected according to their suggestions. As always, our sincere thanks go to the reviewers – in total, the 57 review statements contained nearly 28,000 words. We are thankful to the members of the program committee for the time and passion they have put into giving useful advices on how to improve the papers.

As a result, the workshop presented a broad view on the recent development in the field of software-intensive business within the selected focal areas. All in all, the program committee highlighted four themes across the papers: *startups*, *new product development*, *business models* and *ecosystems*. These categorizations give an interesting vantage point to the ongoing debate in the field of software-intensive business. Our discussion generally focuses on new venture creation, value creation and value capture — themes much researched but still lacking in software specific explanations. This set the tone of this year's workshop.

Software startups. The workshop day was opened by Jason Grendus with his keynote presentation titled *"Business Angel Mindset"*. In his presentation, Grendus shared his experiences on working with, and as one of angel investors.

Klotins [2] continued the theme by asking the question *"...[A]re start-ups snowflakes?"* The work made an effort to understand if, and to which extent, software startups are unique in how they approach software engineering. Klotins [2] took a narrow view of engineering; we can of course ask the same question more broadly: How unique are software-intensive business and to which extent it warrants its own research. Much of the workshop actually focused on just this, highlighting the software specific undertones in the broader understanding about business, management, and engineering.

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

Kemell et al. [3] showed through a grey literature review over a hundred metrics for software startups. Many of these make sense in a broader analysis of business and engineering, but some show clear specificity on understanding software-intensive business. Khanna and Wang [4] presented a framework on how startup teams could utilize experience-based learning in their work.

Software-intensive business, particularly through platforms, is a low entry barrier business driven by minimum viable product experimentation. This has many research implications. It highlights the importance of education on new venture creation. Gutbrod and Münch [5] look at teaching lean startup principles and in particularly how entrepreneurs can identify core assumptions in a fast-paced business environment. In the same line, Chanin et al. [6] looked at how software startup education impacts the success or failure of startups.

Yrjönkoski and Suominen [7] studied effectuation as a frame for network decisions in a software startup. Their results show that effectuation behaviour might be an useful approach for managers in the early fuzzy phases of a startup. In addition, they point out avenues for further work on the concept of effectuation.

New product development. The field of software-intensive business is tightly intertwined with the actual development of software artifacts. To foster the development of the field, Schönhofen et al. [8] address in their paper how the ISO 16355 standard can be used to support the development of the software-intensive field. The standard, which is based on Quality Function Deployment (QFD), seems a promising starting point for future work.

With a similar focus, Saltan et al. [9] identified new research directions for software product management based on a case study focusing on five software-intensive companies. Melegati & Wang [10] focused their understanding on how software startups innovate in the dynamic market, finding that literature did not differentiate startups based on the innovations they develop, leaving much for further research to uncover. Münch et al. [11] addressed, in their study, the problems of the traditional roadmapping in a software-intensive company. They conclude that the traditional approach is not suitable anymore in agile and innovative environments.

Bosch et al. [12] identified, from their empirical material, three different approaches to the software development: Requirement-driven, Outcome-driven and AI-driven development approaches. Their study presents a new and interesting way to characterize software development work in companies. In addition, Bosch et al. [12] provided a framework for deciding when and what approach to use.

Business models. As the field of software-intensive business aims to cross the gap between technology and business, also business models were strongly present in the workshop. Spijkerman and Jansen [13] presented a survey on ten open-source software companies' business models and summarize their key findings as an open-source software business model blueprint.

Saltan and Seffah [14] presented a mapping study for identifying the technical and business challenges of SaaS adoptions. As a result, they present a framework for identifying the challenges and required a formation of a research agenda.

Yrjönkoski [15] continued the same line of research by surveying literature on how small-and-medium-sized enterprises (SMEs) should organize the transformation from on-premise products to SaaS solutions. His results show, that while results from large enterprises with enough time and resources have been presented, there is a lack of work reporting how SMEs have carried out the transformation with fewer resources.

Ecosystems & platforms. Also software ecosystems and platforms were presented in the workshop. Molenaar et al. [16] studied how partners perceive the keystone's power in a software ecosystem. Their study reveals new insights in partner-keystone dynamics in the software ecosystems. Joelsson et al. [17] continued the same approach and went on to highlight the active role of *prosumers* — "users who consume as well as produce" — in the studied software ecosystem. This type of actors is specific to digital environments and their role is understudied.

Petrik and Herzwurm [18] studied industrial Internet of Things (iIoT) platforms and based on the interviews, they present a business model taxonomy for iIoT platforms. Hajikhani [19] focused on social media platforms and entrepreneurial discussions in smart cities. By focusing on the case of London, his study advances our understanding on social media's impact on an innovation and entrepreneurial ecosystem.

Turunen and Mäntymäki [20] observed a lack of understanding of psychosocial dynamics in ecosystem studies. They use the concept of *collective consciousness* as a tool for characterizing the ecosystems as complex networks of heterogeneous actors. Their study works as an interesting opening to ecosystem scholars to widen the approaches used to understand the complex phenomenon.

December 2018

Sami Hyrynsalmi, Arho Suominen,
Christopher Jud, Xiaofeng Wang,
Jan Bosch & Jürgen Münch

References

1. Abrahamsson, P., Bosch, J., Brinkkemper, S., Mädche, A.: Software Business, Platforms, and Ecosystems: Fundamentals of Software Production Research (Dagstuhl Seminar 18182). Dagstuhl Reports **8**(4) (2018) 164–198
2. Klotins, E.: Software start-ups through an empirical lens: are start-ups snowflakes? In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 1–14
3. Kemell, K.K., Wang, X., Nguyen-Duc, A., Grendus, J., Tuunanen, T., Abrahamsson, P.: 100+ metrics for software startups – a multi-vocal literature review. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 15–29
4. Khanna, D., Wang, X.: The buried presence of entrepreneurial experience-based learning in software startups. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 30–39
5. Gutbrod, M., Münch, J.: Teaching lean startup principles: An empirical study on assumption prioritization. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 245–253
6. Chanin, R., Khanna, D., Kemell, K.K., Xiaofeng, W., Sales, A., Prikladnicki, R., Abrahamsson, P.: Software startup education around the world: A preliminary analysis. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 219–229
7. Yrjönkoski, K., Suominen, A.: Effectuation as a frame for networking decisions – the case of a Finnish information technology start-up. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 230–244
8. Schönhofen, F., Schockert, S., Herzwurm, G.: ISO 16355 in software-intensive business. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 158–163
9. Saltan, A., Jansen, S., Smolander, K.: Decision-making in software product management: Identifying research directions from practice. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 164–176
10. Melegati, J., Wang, X.: Do software startups innovate in the same way? a case survey study. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 193–201
11. Münch, J., Trieflinger, S., Lang, D.: Why feature-based roadmaps fail in rapidly changing markets: A qualitative survey. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 202–218
12. Bosch, J., Olsson, H.H., Crnkovic, I.: It takes three to tango: Requirement, outcome/data, and AI driven development. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 177–192
13. Spijkerman, Z., Jansen, S.: The open source software business model blueprint: A comparative analysis of 10 open source companies. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 128–145

14. Saltan, A., Seffah, A.: Engineering and business aspects of SaaS model adoption: Insights from a mapping study. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 115–127
15. Yrjönkoski, T.: How to support transformation from on-premise products to saas? position paper for future research. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 144–157
16. Molenaar, S., van Vliet, M., Beelen, L., Jansen, S.: Business as usual? on the nature of relationships in enterprise software platform ecosystems. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 40–56
17. Joelsson, T.N., Hyrynsalmi, S., Molenaar, S.: The role of prosumers in the evolution of a software ecosystem: Case Steam. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 72–88
18. Petrik, D., Herzwurm, G.: Platform ecosystems for the industrial internet of things – a software intensive business perspective. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 57–71
19. Hajikhani, A.: Entrepreneurial oriented discussions in smart cities: Perspectives driven from systematic use of social network services data. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 89–103
20. Turunen, M., Mäntymäki, M.: Collective consciousness in business ecosystems. In: Proceedings of the International Workshop on Software-intensive Business: Start-ups, Ecosystems and Platforms 2018 (SiBW 2018). (2018) 105–114

Software start-ups through an empirical lens: are start-ups snowflakes?

Eriks Klotins

Blekinge Institute of Technology, Karlskrona, Sweden
eriks.klotins@bth.se

Abstract. Most of the existing research assume that software start-ups are "unique" and require a special approach to software engineering. The uniqueness of start-ups is often justified by the scarcity of resources, time pressure, little operating history, and focus on innovation. As a consequence, most research on software start-ups concentrate on exploring the start-up context and are overlooking the potential of transferring the best engineering practices from other contexts to start-ups.
In this paper, we examine results from an earlier mapping study reporting frequently used terms in literature used to characterize start-ups. We analyze how much empirical evidence support each characteristic, and how unique each characteristic is in the context of innovative, market-driven, software-intensive product development.
Our findings suggest that many of the terms used to describe start-ups originate from anecdotal evidence and have little empirical backing. Therefore, there is a potential to revise the original start-up characterization.
In conclusion, we identify three potential research avenues for further work: a) considering shareholder perspective in product decisions, b) providing support for software engineering in rapidly growing organizations, and c) focusing on transferring the best engineering practices from other contexts to start-ups.

Keywords: start-ups · software engineering · engineering context

1 Introduction

In recent years, software start-ups have gained attention from the research community. In 2014, a systematic mapping study by Paternoster et al. [39] highlighted the lack of relevant research addressing software engineering in start-ups. Results of this paper are reused by most subsequent studies on software start-ups.

In 2016, Unterkalmsteiner et al. [52] published a research agenda identifying further research directions in the area. These directions explore start-ups from software engineering perspective and only superficially touches upon other, e.g. marketing and business, aspects of start-ups. The underlying idea is that the core of a start-up is development and maintenance of a software-intensive product.

Thus, shortcomings in the product development could hinder any subsequent attempts to build a sustainable business around it [27].

Since 2014, a substantial corpus of empirical data on software start-ups has been collected and analyzed, for example, Giardino et al. [17], Klotins et al. [30,28], and Tripathi et al. [51]. Several models are proposed to explain software engineering in start-ups, for example, Giardino et al. [8,18] and Klotins et al. [31].

Most of the recent research on software start-ups focus on exploring engineering context and used practices. The exploration is motivated by the premise that start-ups are "special" and "unique", thus require a special approach to software engineering, for example, Sutton [49], Blank [5], Gralha et al. [19], and Duc et al. [13]. At the same time, systematic adoption of existing engineering practices for use in start-ups had attracted little attention [29,4].

The empirical data, for instance, Coleman et al. [10], Klotins et al. [28,30] and Giardino et al. [17], show little evidence of anything special, regarding software engineering, in start-ups compared to other market-driven organizations developing innovative software-intensive products. Such results invite to revisit the initial premise.

Understanding to what extent software start-ups are different from established organizations is central to transferring the best engineering practices from other contexts to start-ups. If start-ups are different, the differences need to be explored to develop start-up specific engineering practices. If start-ups are not different, further research needs to emphasize the transfer of the best engineering practices from other contexts to start-ups.

There has been a limited success with formulating a crisp and distinctive definition of a software start-up [44,52]. Ries [43] broadly defines start-ups as human institutions aiming to deliver new products or services under extreme uncertainty. Carmel [6] defines start-ups as new, market-driven companies aiming to launch software product fast with minimal resources. Unterkalmsteiner et al. [52] define software start-ups as newly founded companies developing software-intensive products under time and resource pressures. In our earlier study, we define start-ups as small companies created to develop and to market an innovative and software-intensive product and to aim to benefit from economies of scale [28]. These definitions describe software start-ups, however miss to convey any distinctive features.

Blank [5] argues the key difference between start-ups and established organizations is that established organizations aim to execute their business model, while start-ups are searching for one. To software engineers, this difference translates into a focus on iterative development, frequent product releases, and extensive use of customer feedback. A very similar approach is used for market-driven product development in established organizations [12].

Paternoster et al. [39] compile a list of recurring terms describing software start-ups. The terms are, for example, lack of resources and experience, time pressure, small team, high risk of failure among others. This list is often used by later studies, for example, Gralha et al. [19], Giardino et al. [17,8], and Klotins et al. [30], to define what is a start-up and to justify their uniqueness. However, the list is meant to "illustrate how authors use the term software startup", and does not imply any empirical grounding.

The objective of this study is to examine how much empirical support there is for "unique" characteristics of start-ups. We analyze the list of start-up characteristics proposed by Paternoster et al. [39] and trace the supporting literature. Then, we examine the literature to estimate how much empirical support there is for each characteristic.

The rest of this paper is structured as follows: In Section 2 we examine the terms and the supporting evidence. In Section 3 we discuss our findings. Section 3 concludes the paper.

2 Start-up characteristics

We use the list of recurring terms characterizing software start-ups by Paternoster et al. [39], to drive our analysis. The original list contains the following characteristics:

1. Lack of resources - Economical, human, and physical resources are extremely limited.
2. Highly Reactive - Startups are able to quickly react to changes of the underlying market, technologies, and product (compared to more established companies)
3. Innovation - Given the highly competitive ecosystem, startups need to focus on highly innovative segments of the market.
4. Uncertainty - Startups deal with a highly uncertain ecosystem under different perspectives: market, product features, competition, people and finance.
5. Rapidly Evolving - Successful startups aim to grow and scale rapidly.
6. Time-pressure - The environment often forces startups to release fast and to work under constant pressure (terms sheets, demo days, investors' requests)
7. Third party dependency - Due to lack of resources, to build their product, startups heavily rely on external solutions: External APIs, Open Source Software, outsourcing, COTS, etc.
8. Small Team - Startups start with a small numbers of individuals.
9. One product - Company's activities gravitate around one product/service only.
10. Low-experienced team - A good part of the development team is formed by people with less than 5 years of experience and often recently graduated students.
11. New company - The company has been recently created.
12. Flat organization - Startups are usually founders-centric and everyone in the company has big responsibilities, with no need of high-management.
13. Highly Risky - The failure rate of startups is extremely high.
14. Not self-sustained - Especially in the early stage, startups need external funding to sustain their activities (Venture Capitalist, Angel Investments, Personal Funds, etc.).

15. Little working history - The basis of an organizational culture is not present initially.

For the brevity of our discussion, we group these terms them into 6 categories as some of the terms appear to be related.

In the following subsections, we examine sources of the characteristics. We look into papers, identified by the review [39], to find empirical support each start-up characteristic. In our review, we include both papers listed by the mapping study and any relevant papers referenced by the listed papers. In essence, we attempt to trace the original statement, a piece of data that inspired the formulation of each characteristic. In addition, we discuss to what extent each characteristic is relevant in other types of organizations.

2.1 Lack of resources and dependency on external sponsors

Lack of human, economic, and physical resources to support product engineering is the most frequently used term to describe software start-ups. It is related to dependencies on 3rd parties for funding and having not enough cash-flow to be self sustainable [39].

Following the references, we found 24 papers, of which 17 analyze empirical data. We review these 17 papers to understand what exact empirical data was the basis for claiming that lack of resources and dependency of external sponsors are characteristic to start-ups.

Some of the papers discuss the need or intention to allocate resources to support product engineering, and not the lack of resources as a challenge [55,54,7]. Coleman et al. [11] reference an experience report from a start-up company. The start-up, operating in 1992 was not able to afford then costly Internet connection and had relied on public Internet access elsewhere. May [34] discusses wasted resources in a start-up due to poor work ethics and using sub-optimal technologies. Mudambi et al. [36] and Yoo et al. [56] argue that small organizations have lesser resources at hand than larger organizations and may not yet have a sustainable revenue, thus resource allocation is an ongoing issue. Later studies elaborate on the impact of resource shortages.

Giardino et al. [17] report allocation of resources as one of the Top 10 challenges in start-ups, and elaborates that a studied company was unable to solve some technical problems in the product due to insufficient resources. Lindgren et al. [33] report that start-ups were not able to utilize experimentation to a full extent due to limited resources. Jorgensen [24] report that shortages in human resources caused delays in product development, and a project was canceled due to an insufficient budget.

Related work on project resource management suggests securing sufficient resources is one of the critical steps in project inception and is linked to project success [47]. In both plan-driven and agile environments, the presence of a committed sponsor is one of the key denominators for project success [9]. The trade-offs between features, resources, and quality, are common in any project [25,14]. In this aspect, start-ups do not look any different.

A study investigating the impact on venture capital to start-ups prospects found that external funding has no significant effect on start-up outcome [48]. Therefore, the focus of further research and practice should be on better methods for engineering resource planning, control, and risk management, to make the best use of any amount of resources. Hadley et al. [20] presents similar findings suggesting an association between venture capital and negative long-term consequences.

A report by Harvard Business Review [38] report that venture capitalists prefer investing in start-ups with younger founders, even though the odds of commercial success are with older and more experienced founders. The report points out that younger founders could be more financially constrained, thus be more willing to cede their business to venture capitalists at a lower price. In other words, young and inexperienced founders could provide higher returns of investment for venture capitalists.

The related work so far does not present any convincing evidence that start-ups would experience the trade-off between resources, scope, and quality differently than other organizations [25]. However, the related work suggests that a potential difference between start-ups and established organizations could be that in an established organization project sponsor and the project team are from the same organization, thus share the same goals to serve customers, improve internal efficiency, and fulfill organization's mission. However, start-ups are often funded by other organizations, for example, venture capitalists. Thus, their goals may not always be aligned [48,38].

As shown by Azoulay et al. [38], venture capitalists could aim to maximize their return on investment. Start-up founders, in turn, could be motivated by an intent to bring their ideas to market, desire for autonomy, and need for accomplishment among other factors [15].

2.2 Time pressure

Time pressure is often used in combination with a lack of resources to describe start-ups [37]. The pressure supposedly originates from investors, external deadlines, and contracts. Following the references, we found 13 supporting papers, of which 6 use empirical data [39].

Examining the papers closer, we found that none of the papers use any data to justify the time pressure in start-ups. However, the papers present a discussion motivating the need for faster delivery time to reduce opportunity cost [5,50,40]. Start-ups aim to spend as little time as possible on activities that have an uncertain contribution to customer value, e.g., building invented features.

Giardino et al. [8] identifies development speed as the core concept in start-ups. It is motivated by the need to keep the team's morale high and to validate the product idea fast. Another study by Giardino et al. [17] links time pressure with available resources and the need to establish a sustainable stream of revenue quickly.

These findings suggest that the time pressure originates from internal considerations and resource limitations, and not from competition or external deadlines. Thus, start-ups may have relative freedom to control the development pace and address the trade-off between quality and speed. Concerning time pressure, established companies face the same opportunity costs. However, they may have more resources at hand to sponsor the product development for longer.

2.3 Innovation

Focus on innovative technologies, products, and market segments is another term used to characterize start-ups [39]. Following the references, we identified in 15 papers, of which 9 uses empirical data, concerning innovation in start-ups.

These studies show that start-ups use innovative offerings primarily to differentiate from other competitors in the market [55,7]. The innovation in start-ups is to a large extent incremental and adds slight improvements to an existing product [32,55]. The innovative aspects can concern product features, quality, packaging, and marketing [26].

Continuous innovation, driven by the innovation strategy, is essential to maintain a competitive edge [32,26]. Heitlager et al. [22] argue, albeit without empirical support, that start-ups start with product innovation to enter the market, followed by process innovation to improve efficiency.

Multi-vocal literature recognizes multiple types of innovation, for example, incremental and process innovation, business model innovation, radical and disruptive innovation [1]. Incremental, process and business model innovation appears to be most suited for small organizations as they focus on improving already known features, activities, and business models [26]. However, disruptive and radical innovation requires substantial investments and time to replace existing products and create entirely new markets with new business models [2]. Thus, these types of innovation could be less suited for resource-strapped start-ups.

Regarding innovation, larger organizations may have the leverage to push more ambitious innovations than small start-ups. For example, Apple had created several disruptive innovations by launching its music platform, iPhone, and AppStore. Such innovations were enabled by their experience within the market, human, organizational and economic resources, and their brand name [53]. However, start-ups lack most, if not all, such enablers. Regarding innovation, start-ups may have to be more modest than established organizations [46].

2.4 Rapidly evolving new company

Terms such as rapid evolution, a new company, small and flat team, focus on one product, and little working history are supported by 34 papers, 22 of them analyzing empirical data [39].

There is an agreement among the papers that start-ups are new organizations established by one or a few founders championing the product idea [55,7,34].

More people, resources, and processes are brought in to support product development and customer service. More processes and artifacts are introduced as the organization grows [7,10].

Surprisingly, none of the studies present data illustrating the start-up growth. The growth is extrapolated from interviewee reflections (e.g. Coleman et al. [11]), a generalized model (Carmel. [7]), and plans to grow customer volume and market share rapidly (Yogendra et al. [55]).

Later studies identify evolving engineering practices in start-ups. Gralha et al. [19] and Melegati et al. [35] identify that requirements engineering practices in start-ups develop from informal to more structured as the start-up matures. Giardino et al. [8] identifies a similar pattern in the adoption of agile practices. Early on, start-ups opt for an ad-hoc approach to engineering and introduce new practices as needed. Introduction of new practices and processes impair development speed, however improve coordination and product quality [28,8].

Established organizations, compared to start-ups, are per definition more stable. Although organizational changes occur in established organizations, they are supported by processes, infrastructure, and concern one or few aspects of the organization at the time [58]. Therefore, rapid evolution in multiple aspects at once could be the most substantial difference between start-ups and other types of organizations.

2.5 Lack of experience

Inexperienced start-up teams are reported as a common theme in literature [39]. This term is supported by 7 papers. However, by looking at the papers closer, we found that none of them present any empirical data supporting the statement.

By analyzing the papers, we found several studies presenting data and analysis providing a strong link between the experience of the teams and prospects of start-up success [26,56,7]. More experienced people require less management [10], and are an essential resource for rapid product development [7,3]. However, May [34] and Giardino et al. [17] note that it is not always easy to find skilled and motivated individuals.

A report by Harvard Business Review [38] analyzing a large sample of founders from the US show that most start-up founders are 30 - 50 years old. The average age of commercially successful start-up founder is 45. Authors of the report emphasize the importance of previous experience and acumen to start a new business that comes with older age. Such findings refute the idea of young and inexperienced start-up founders as a typical case.

Other studies add further support for the importance of technical and business experience to start-up success [57,41]. Giardino et al. [8] emphasizes the importance of a small and motivated team of skilled individuals. However, we could not find any evidence that start-ups would have disproportionally more inexperienced engineers than any other type of organization.

Established organizations put substantial effort into on-boarding new software engineers. For example, by providing on-the-job training, mentoring, employee guides, and so on [23]. It could take several months until a recruit reaches full productivity [16]. A small start-up may lack the capacity to provide such resources to new engineers. As a consequence, start-ups may aim to hire engineers with relevant technical and domain knowledge to compensate for the lack of on-the-job training.

2.6 Highly risky

High risk of failure and uncertainty is identified as characteristic to start-ups is supported 12 studies, of which 8 uses empirical data [39].

Examining the studies further, we found that none of them present any data on start-up failure rate. Blank [5] estimates a 75% failure rate among start-ups and motivates it by a report from Harvard Business School. However, we were not able to find the original report.

Looking further, we found a study reporting small business survival rate of 66% after the first year, and 40% after six years or more [21]. The sample includes all types of recently established small businesses. While exact numbers from different sources vary, they agree that most new companies do not survive past the first few years. That said, we were not able to find any credible source estimating a general failure rate among start-ups.

Carmel [6] emphasizes that launching a new venture is inherently associated with the risk of failure. However, estimating success and failure rate of start-ups is difficult. Likely, many start-up initiatives are closed down before they appear on any records. After closure, there is no evidence left behind to be studied. Part of the difficulty to estimate start-up failure rate is lack of a clear definition of what is a start-up, and what are their success and failure conditions.

Traditional project management literature considers a project successful if it is delivered within budget, time, and scope [47]. The economic perspective on start-ups identifies return of investment as the accurate measure of success [42]. Customer-centric view proposes to use customer satisfaction to assess the project success [45]. Carmel [6] argue that speed is the essential success metric in start-ups.

So far, the related work does not present any evidence that start-ups would have substantially different survival rate than other types of recently established ventures. However, as we have discussed earlier, start-ups may have stakeholders with different interpretations of success. For example, the investors could be looking for specific return of investment ratio. The odds of attaining such specific objectives could be much lower than of general survival of the company.

3 Discussion

We perform this inquiry to understand if there is enough evidence to claim that start-ups are different from established companies and need a different approach to software engineering. We examine 15 start-up characteristics that are often used to define and differentiate start-ups from established organizations.

By reviewing the literature, we identify several common shortcomings. Firstly, many studies present an anecdotal characterization of start-ups. Such characterization of start-ups is often placed in the introduction, motivating the study. Meanwhile, the research itself focuses on different aspects that neither add or remove support for the characteristics. Such anecdotes propagate, are generalized by further studies, and cause misconceptions about engineering start-ups.

Secondly, studies investigating start-ups rarely, if at all, discuss their findings in a broader context. As a consequence, some challenges, for example, lack of resources and innovation, are presented as unique to start-ups. Such narrow focus takes away the opportunity to transfer the best engineering practices from other contexts to start-ups, and vice versa.

By evaluating the actual empirical evidence, we find little support for most of the characteristics. For example, we could not find any empirical evidence showing that start-up teams are inexperienced. Quite the opposite, empirical studies show that start-ups are often founded by middle-aged entrepreneurs with substantial experience and business acumen. Furthermore, some of the characteristics that are presented as "unique" to start-ups are common in other types of organizations. For example, the challenge of balancing project scope with available resources is hardly unique to start-ups. In other words, by examining the literature, we could not find convincing empirical evidence that start-ups would be in any way "unique" regarding software engineering. Such results suggest that the focus of further research should be on transferring the best engineering practices from established organizations to start-ups.

We identify several limitations concerning our study. The start-up characteristics discussed in this paper are based on work by Paternoster et al. [39]. There could be other studies more accurately describing start-ups and emphasizing their distinctive characteristics. However, to our best knowledge, the terms identified by Paternoster et al. are the most commonly used, thus serve as a good enough basis to raise the discussion on what is so special about software engineering in start-ups.

The literature analyzed in this paper is identified by following traceability information provided by Paternoster et al. [39]. There is a threat that this information is incomplete and we may have overlooked some important studies. To address this treat, and explore a concept in a broader context, we perform independent searches for relevant literature.

Our discussion is limited only to software engineering perspective of start-ups. Other perspectives, for example, business, finances, and marketing could present more distinct differences between start-ups and established organizations. Such other perspectives are left out from our discussion.

4 Conclusions and further work

In this paper, we examine the commonly used characteristics to distinguish between start-ups and established organizations. We found that most of the frequently used start-up characteristics have little empirical support, and some of the characteristics are present in larger organizations as well. We conclude that the terms characterizing software start-ups, and the definition of software start-ups from software engineering perspective need to be revised.

Such finding has implications to our main question whether or not start-ups are special, and should use different engineering practices than small-medium enterprises and other types of organizations. We could not find convincing evidence that start-ups need a different approach to engineering than other types of organizations. We found that rapid evolution and conflicting stakeholders objectives could be adding extra complexity to software engineering. Such additional complexity suggests that start-ups should be more, not less, structured in following the best engineering practices.

From our analysis, we identify three potential research directions concerning software start-ups.

1. Rapid evolution: Growing an organization from a few people to multiple teams working together in a short time requires an evolution of communication and coordination practices as well. Practices that work with few engineers, customers, and a small product, will not suffice in a larger team, thousands of customers and a complex product. There are plenty of engineering practices aimed at dynamic environments, e.g., agile. However, realizing the need for, selection, and continuous adoption of new practices is a major engineering challenge.

2. Thinner margins of error: Given their small size and dependency on external sponsors, start-ups have little margin for errors. The errors may concern both product decisions, e.g., what features and quality to build, and process decisions, e.g., determining the most efficient way of delivering the features. Larger organizations could cover losses of one product with profits from another. And, compensate for inefficient practices with more resources. However, in start-ups failure to deliver customer value quickly usually means the closure of the company. To software engineers, this translates into the need for proven engineering methods, continuous process improvement, stricter control over resource utilization, and better risk management.

3. Misaligned stakeholder objectives: When project sponsors and the project team are from the same organization, they share the same high-level goals, e.g., to serve their customers, and fulfill the company's mission. However, in start-ups project sponsors could be from a different organization, thus may have very different goals. For instance, venture capitalists may aim to maximize the returns of investment, while a start-up could aim to pioneer an innovative technology. To software engineers, this implies the need to balance the interests of different stakeholder groups, namely, customers, shareholders, and the start-up itself.

5 Acknowledgments

Ideas presented in this paper arise from discussions with Dr. M. Unterkalmsteiner, Prof. Dr. T. Gorschek, and members of Software Start-up Research Network[1].

The author would like to thank all the reviewers for their time and valuable input in shaping this paper.

References

1. 15 types of innovation. https://thegentleartofsmartstealing.wordpress.com/types-of-innovation/, accessed: 2018-10-05
2. Radical innovation. https://www.innovation-creativity.com/radical-innovation.html, accessed: 2018-10-05
3. Ambler, S.: Lessons in agility from Internet-based development. Software, IEEE **19**(2), 66 – 73 (2002)
4. Berg, V., Birkeland, J., Nguyen-Duc, A., Pappas, I., Jaccheri, L.: Software startup engineering: A systematic mapping study. Journal of Systems and Software (2018)
5. Blank, S.: Why the Lean Start Up Changes Everything. Harvard Business Review **91**(5), 64 (2013)
6. Carmel, E.: Rapid development in software package startups. In: Proc. 27th Hawaii Int'l Conf. System Sciences (1994)
7. Carmel, E.: Time-to-completion in software package startups. In: 1994 Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences (1994)
8. Carmine, G., Paternoster, N., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software Development in Startup Companies: The Greenfield Startup Model. IEEE Transactions on Software Engineering **X**(September), 233 (2016)
9. Chow, T., Cao, D.B.: A survey study of critical success factors in agile software projects. Journal of Systems and Software **81**(6), 961–971 (2008)
10. Coleman, G., O'Connor, R.V.: An investigation into software development process formation in software start-ups. Journal of Enterprise Information Management **21**(6), 633–648 (2008)
11. Coleman, G., O'Connor, R.V.: An investigation into software development process formation in software start-ups. Journal of Enterprise Information Management **21**(6), 633–648 (2008)
12. Dahlstedt, Å.G., Karlsson, L., Persson, A., Natt och Dag, J., Regnell, B.: Market-Driven Requirements Engineering Processes for Software Products - a Report on Current Practices. In: International Workshop on COTS and Product Software, RECOTS 2003 (2003)
13. Duc, A.N., Dahle, Y., Steinert, M., Abrahamsson, P.: Towards understanding startup product development as effectual entrepreneurial behaviors. In: International Conference of Software Business. pp. 199–204. Springer (2017)
14. Elonen, S., Artto, K.A.: Problems in managing internal development projects in multi-project environments. International journal of project management **21**(6), 395–402 (2003)
15. Estay, C., Durrieu, F., Akhter, M.: Entrepreneurship: From motivation to start-up. Journal of International Entrepreneurship **11**(3), 243–267 (2013)

[1] Software Start-up Research Network: http://softwarestartups.org

16. Ganzel, R.: Putting out the welcome mat. Training **35**(3), 54–60 (1998)
17. Giardino, C., Bajwa, S.S., Wang, X.: Key Challenges in Early-Stage Software Startups. In: Agile Processes, in Software Engineering, and Extreme Programming. vol. 212, pp. 52–63 (2015)
18. Giardino, C., Wang, X., Abrahamsson, P.: Why Early-Stage Software Startups Fail : A Behavioral Framework pp. 27–41 (2014)
19. Gralha, C., Damian, D., Wasserman, A.I.T., Goulão, M., Araújo, J.: The evolution of requirements practices in software startups pp. 823–833 (2018)
20. Hadley, B., Gloor, P.A., Woerner, S.L., Zhou, Y.: Analyzing vc influence on startup success: They might not be good for you
21. Headd, B.: Redefining business success: Distinguishing between closure and failure. Small business economics **21**(1), 51–61 (2003)
22. Heitlager, I., Helms, R., Brinkkemper, S.: A tentative technique for the study and planning of co-evolution in product. In: Software Evolvability, 2007 Third International IEEE Workshop on. pp. 42–47. IEEE (2007)
23. Johnson, M., Senges, M.: Learning to be a programmer in a complex organization: A case study on practice-based learning during the onboarding process at google. Journal of Workplace Learning **22**(3), 180–194 (2010)
24. Jørgensen, K.: Newcomers in a global industry: Challenges of a norwegian game company. Games and Culture p. 1555412017723265 (2017)
25. Junk, W.S.: The Dynamic Balance Between Cost, Schedule, Features, and Quality in Software Development Projects. Computer Science Dept., University of Idaho **SEPM-001** (2000)
26. Kakati, M.: Success criteria in high-tech new ventures. Technovation **23**(5), 447–457 (may 2003)
27. Klotins, E., Unterkalmsteiner, M., Gorschek, T.: Software Engineering Antipatterns in start-ups. In review by IEEE Software (2017)
28. Klotins, E., Unterkalmsteiner, M., Catzipetrou, P., Gorschek, T., Prikladnicki, R., Tripathi, N., Pompermaier, L.B.: A progression model of software engineering goals , challenges , and practices in start-ups. Transactions of Software Engineering **xxx**(9), 1–22 (2018)
29. Klotins, E., Unterkalmsteiner, M., Gorschek, T.: Software Engineering Knowledge Areas in Startup Companies : a mapping study. In: Lecture Notes in Business Information Processing. pp. 245–257. Springer (2015)
30. Klotins, E., Unterkalmsteiner, M., Gorschek, T.: Software engineering in start-up companies: An analysis of 88 experience reports. Empirical Software Engineering pp. 1–35 (2018)
31. Klotins, E., Unterkalmsteiner, M., Gorschek, T.: Software-intensive product engineering in start-ups: a taxonomy. IEEE Software **35**(4) (2018)
32. Linda, S.l.L.: Chinese entrepreneurship in the internet age: Lessons from alibaba. com. International Science Index, Economics and Management Engineering **4**(12) (2010)
33. Lindgren, E., Münch, J.: Raising the odds of success: the current state of experimentation in product development. Information and Software Technology **77**, 80–91 (2016)
34. 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: Agile Conference. pp. 141–147. Ieee (aug 2012)
35. Melegati, J., Goldman, A., Paulo, S.: Requirements Engineering in Software Startups : a Grounded Theory approach. In: 2nd International Workshop on Software Startups, Trondheim, Norway (2016)

36. Mudambi, R., Treichel, M.Z.: Cash crisis in newly public internet-based firms: An empirical analysis. Journal of Business Venturing **20**(4), 543–571 (2005)
37. Nguyen-Duc, A., Wang, X., Abrahamsson, P.: What influences the speed of prototyping? an empirical investigation of twenty software startups. In: International Conference on Agile Software Development. pp. 20–36. Springer (2017)
38. P. Azoulay, B. Jones, J.D.K.J.M.: Research: The average age of a successful startup founder is 45. Harvard Business Review (Jul 2018)
39. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: A systematic mapping study. Information and Software Technology **56**(10), 1200–1218 (oct 2014)
40. Payne, J.W., Bettman, J.R., Luce, M.F.: When time is money: Decision behavior under opportunity-cost time pressure. Organizational behavior and human decision processes **66**(2), 131–152 (1996)
41. Politis, D.: Does prior start-up experience matter for entrepreneurs' learning? a comparison between novice and habitual entrepreneurs. Journal of small business and Enterprise Development **15**(3), 472–489 (2008)
42. Reid, G.C., Smith, J.A.: What makes a new business start-up successful? Small Business Economics **14**(3), 165–182 (2000)
43. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown busi edn. (2011)
44. Sánchez-Gordón, M.L., Colomo-Palacios, R., de Amescua Seco, A., O'Connor, R.V.: The route to software process improvement in small-and medium-sized enterprises. In: Managing Software Process Evolution, pp. 109–136. Springer (2016)
45. Sauvola, T., Lwakatare, L.E., Karvonen, T., Kuvaja, P., Olsson, H.H., Bosch, J., Oivo, M.: Towards customer-centric software development: a multiple-case study. In: Software Engineering and Advanced Applications (SEAA), 2015 41st Euromicro Conference on. pp. 9–17. IEEE (2015)
46. Slinger Jansen, Sjaak Brinkkemper, Ivo Hunink, C.D.: Pragmatic and Opportunistic Reuse in Innovative Start-up Companies. IEEE Software pp. 42–49 (2008)
47. Snyder, C.S.: A guide to the project management body of knowledge: Pmbok (®) guide. Project Management Institute: Newtown Square, PA, USA (2014)
48. Suominen, A., Hyrynsalmi, S., Still, K., Aarikka-stenroos, L.: Software Start-up failure An exploratory study on the impact of investment pp. 55–64 (2017)
49. Sutton, S.M., Cubed, E.C., Andretti, M.: The Role of Process in a Software Start-up. IEEE Software **17**(4), 33–39 (2000)
50. Tingling, P., Saeed, A.: Extreme programming in action: a longitudinal case study. In: HCI International. pp. 242–251 (2007)
51. Tripathi, N., Klotins, E., Prikladnicki, R., Oivo, M., Pompermaier, L.B., Kudakacheril, A.S., Unterkalmsteiner, M., Liukkunen, K., Gorschek, T.: An anatomy of requirements engineering in software startups using multi-vocal literature and case survey. Journal of Systems and Software (2018)
52. Unterkalmsteiner, M., Abrahamsson, P., Wang, X., Nguyen-Duc, A., Shah, S., Bajwa, S.S., Baltes, G.H., Conboy, K., Cullina, E., Dennehy, D., et al.: Software startups–a research agenda. e-Informatica Software Engineering Journal **10**(1) (2016)
53. West, J., Mace, M.: Entering a mature industry through innovation: Apple's iphone strategy. In: DRUID Summer Conference. pp. 18–20 (2007)
54. Yoffie, D.B., Cusumano, M.A.: Building a company on internet time: Lessons from netscape. California Management Review **41**(3), 8–28 (1999)

55. Yogendra, S., Sengupta, S.: Aligning business and technology strategies: a comparison of established and start-up business contexts. In: Engineering Management Conference, 2002. IEMC'02. 2002 IEEE International. vol. 1, pp. 2–7. IEEE (2002)
56. Yoo, C., Yang, D., Kim, H., Heo, E.: Key value drivers of startup companies in the new media industry - the case of online games in korea. Journal of Media Economics **25**(4), 244–260 (2012)
57. Zhang, J.: The advantage of experienced start-up founders in venture capital acquisition: evidence from serial entrepreneurs. Small Business Economics **36**(2), 187–208 (2011)
58. Zhou, K.Z., David, K.T., Li, J.J.: Organizational changes in emerging economies: Drivers and consequences. Journal of International Business Studies **37**(2), 248–263 (2006)

100+ Metrics for Software Startups – A Multi-Vocal Literature Review

Kai-Kristian Kemell[1], Xiaofeng Wang[2], Anh Nguyen-Duc[3], Jason Grendus[4], Tuure Tuunanen[1], and Pekka Abrahamsson[1]

[1] University of Jyväskylä, 40014 Jyväskylä, Finland
{kai-kristian.o.kemell, pekka.abrahamsson, tuure.tuunanen}jyu.fi
[2] Free University of Bozen-Bolzano, 39100 Bozen-Bolzano, Italy
xiaofeng.wang@unibz.it
[3] University of Southeast Norway, Norway
angu@usn.no
[4] 3D Ventures Oy, Singapore
jgrendus@gmail.com

Abstract. Metrics can be used by businesses 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 amidst 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 startup are not widely covered within academic literature. To promote research in this area and to create a starting point for it, we have conducted a multi-vocal literature review focusing on practitioner literature in order to compile a list of metrics used by software startups. Said list is intended to serve as a basis for further research in the area, as the metrics in it are based on suggestions made by practitioners and not empirically verified.

Keywords: Software Startup, Metric, Data, Multi-Vocal Literature Review

1 Introduction

The importance of data in business has greatly increased over the last few decades as acquiring, storing, and using it has become both easier and cheaper in the wake of technological progress. This development was further underlined following the still relatively recent emergence of the big data discourse [47], which encouraged organizations to acquire and store vast amounts of data even if they did not necessarily have any present use for it. Data is now often used by various businesses to support decision-making, even though manager intuition is often in practice still just as important in strategic decision-making [26].

For the purpose of decision-making, data can be used in the form of metrics. Metrics are quantifiable measurements of a phenomenon or object. They are present everywhere in our everyday life from measuring height and weight to measuring speed while driving. Even qualitative data can to some extent be made quantifiable with the right approach: a simple yes or no question can be seen as a Boolean of 1 or 0. In terms of quantifying written statements, techniques such as the Likert scale survey, where users rate qualitative statements on a scale of e.g. 1 to 5 based on how much they agree or disagree with them, have been employed.

Much like larger software companies, software startups can also employ various metrics to measure progress and to aid in decision-making. Given that software startups usually operate under a notable lack of resources and in particularly tumultuous contexts [44], software startups can arguably benefit from the use of metrics. Making the right decisions amidst uncertainty can make all the difference between success and failure. However, based on past survey data[1] from 4700 software startups, most of them in fact did not track metrics or did not use the data gained from tracking them to make decisions. More specifically, 41% of these 4700 software startups felt that it was too early for them to track metrics. Out of the remaining 59% of the responses, some 16% did not track metrics either because they did not have the resources to do or because they did not believe it would benefit them, and 14% tracked them but remarked that the data had no influence on their decision-making.

The majority of software startups end in failure [44]. Arguably, the proper use of the right metrics is something that can help alleviate this situation in part. Metrics can alert a business of approaching disasters and give them time to react before the resulting decrease in revenue really hits them. For example, tracking Daily Active Users (DAU) is a metric that gives near real-time data of how a software is doing. If the number suddenly starts dropping dramatically over the course of a few days, something is likely wrong. Perhaps an update was deployed on the day the initial drop started, and perhaps that update dramatically affected the stability of the software on some devices or operating systems. Nonetheless, in a situation where this hypothetical company was not tracking their DAU, this problem may have only become apparent through a dramatic drop in revenue at the end of the month. However, metrics are typically quite context-dependent; for a very early-stage software startup that is still developing their first product and thus has no users yet, tracking the aforementioned DAU serves no purpose.

Though metrics have been extensively studied in various context across disciplines, metrics specifically in relation to software startups is an emerging area of research. While e.g. classic business metrics such as Net Present Value [38] are certainly applicable to software startups as well, our understanding of what metrics are specifically useful for software startups is presently lacking. To this end, we seek to un-

[1] This was a large-scale survey that ultimately collected 10000+ responses, conducted to explore different aspects of software startups. However, after cleaning the data and filtering it based on whether this particular question about metrics was answered, ~4700 responses remained. As the survey was extensive, most questions were not mandatory, and thus not all responses included answers to all of the questions. Additionally, the numbers are approximations as even after cleaning the data of duplicate or dubious responses (e.g. “name: test.com”) no doubt not all of the remaining responses are valid data. Data from the same survey was also used by Wang et al. [48] among others.

derstand what metrics software startups currently use, or are expected to use, based on a multi-vocal literature review focusing primarily on practitioner literature. Through the literature review, we aim to compile an extensive list of potential metrics for software startups, creating fertile ground for further research on metrics in this context. This list is intended to propose potential metrics but offers little insight in which of these metrics *should* be used. Thus, we formulate the research problem of this paper as follows:

RQ: What metrics could software startups use to track progress of their business?

The rest of this paper is structured as follows. In the upcoming second section we discuss software startup metrics as an area of research in relation to extant research across disciplines. In the third section we go over the methodology of this study in detail, and in the fourth section we present our results. The implications and limitations of the results are discussed in the fifth and final section that also concludes this paper.

2 Software Startups and Metrics

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. [48]), 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.

Software Engineering (SE), metrics can be split into process metrics and product metrics [49]. Process metrics are metrics related to the process of creating the software, or maintaining it during its operational life, while product metrics are related to the qualities of the product. Product metrics can be seen to include usability-related metrics as well. Process metrics, on the other hand, account for various method-specific or practice-specific metrics such as lean or agile software development metrics [24]. Website-related metrics can also be considered to be a part of SE metrics, however, as websites are ultimately software [49].

In terms of website metrics specifically, basic metrics related to system (website) performance such as site availability or bandwidth [46] have become less relevant in the wake of technological process, particularly following the popularization of cloud technology. It is now virtually a given that a website can handle any ordinary spikes in traffic load with more capacity being allocated as necessary. Indeed, rather than tracking at system-related metrics, the focus from a business point of view has shifted towards understanding the way users interact with it [4]. While assuring system performance is no less relevant than before, it is now far easier to achieve website stability with modern computational power.

Organizations aim to comprehensively track the way users use their website in order to better understand them and to optimize it accordingly [4]. Generic metrics for this purpose include tracking visit length per page, tracking what the users click (if anything at all), as well as tracking where the users enter the website from. With large amounts of data becoming increasingly cheap and easy to handle, and with tools for gathering and analyzing such data now being readily available (e.g. Google Analytics), tracking individual users in this fashion has become widespread even among smaller organizations, including software startups. This way of tracking users is not limited to websites. Software companies are equally interested in understanding how the users of their software interact with it in practice in order to improve the software based on the data.

Though software startups occasionally also concern themselves with directly studying usability and User Experience (UX), UX and usability are typically evaluated by actively involving users as participants for a study while either directly observing their use or having the users self-report their experiences through a form. Directly confronting users and potential users in order to better understand their needs can be important and is certainly something software startup practitioners often choose to do as well. However, involving users in order to better understand their needs is something that can be carried out in a similar fashion regardless of whether the organization involved is a software startup or a larger organization. We thus consider them to be out of scope for this literature review as the extant studies in the area are already reasonably applicable to the software startup context as well. This is not to say that further studies on UX and usability testing from the point of view of software startups would not be worth carrying out, however.

As for business metrics, conventional business metrics such as the Net Present Value studied in economic disciplines are also applicable to software startup. However, an early-stage software startup may not yet have a single customer or even a product and thus have no revenue, making many of the more conventional financial metrics less relevant to them especially in their earlier stages. Metrics such as Customer Acquisition Cost, which measures the cost of acquiring a new customer by means of e.g. advertising, can be far more useful for such startups. Similarly, software startups aim for explosive growth and highly scalable business models [44] and thus are also likely to be particularly interested in metrics related to growth over shorter periods of time.

Extant research has extensively studied business metrics, website metrics, and software development related metrics [24] in various contexts. On the other hand, academic research specifically focused on metrics from the point of view of software startups is currently scarce. Software startups are to some extent similar to larger software companies and operate within the same area of the software industry. However, software startups also differ from larger or more mature software organizations in various ways. Thus, while conventional business metrics or software metrics not specifically aimed at software startups are likely to be applicable to software startups, they may not be as important to software startups.

Whereas academic literature on metrics from the point of view of software startups is currently scarce, practitioner literature contains various accounts on software startup metrics. In order to promote discussion and to encourage research in the area, we will review some of the practitioner literature in the area and present the practitioners' views on what metrics software startups should utilize. The details of this multi-vocal literature review are discussed next.

3 Methodology

A multi-vocal literature review primarily focusing on practitioner accounts was conducted to collect data for the purpose of formulating a list of preliminary results. As practitioner literature is very heterogeneous in nature, ranging from books to blogs and lacking in common publication platforms such as journals, establishing a fully systematic protocol for reviewing it is challenging due to the vast amount of available data. We nonetheless devised a protocol in order to conduct the review in a semi-systematic fashion. In this case we refer to it as semi-systematic as it consisted of multiple steps, of which the second one was conducted in a systematic fashion.

The literature review consisted of three steps of searching for literature. First, we reviewed popular books written by high-profile practitioner experts (e.g. Eric Ries and Steve Blank) that were relevant from the point of view of metrics. Secondly, we conducted a set of Google searches in order to find less high-profile practitioner literature such as blog posts from various practitioners involved with software startups. Then, using the literature gathered during the first two steps, we finally utilized the snowballing technique to discover more literature discussed in the documents already included for the review.

For the Google searches, we followed a systematic protocol in order to gather higher quality data. The following queries were used for these searches: "software startup metrics", and "startup metrics", "startup metrics list", and "startup what to measure". For each query, the first five pages of results were screened for inclusion. The results were evaluated for inclusion based on the following inclusion criteria:

- The document is not clearly intended as an advertisement for a tool (e.g. a firm writing a blogpost to recommend their own data analytics tool)
- The document presents or discusses specific, actionable metrics (as opposed to non-specific groups of metrics such as sales metrics)
- The document is a textual document and not e.g. a link to a video or a slideshow
- The document is a stand-alone document written under a real name (i.e. not a forum post written under a pseudonym)
- The document is publicly available; not behind a pay-wall or registration
- The document contains metrics that can be employed by most software startups (e.g. not only e-commerce metrics)
- The document is not a duplicate result from another search query

We chose to not limit our inclusions to metrics specifically presented as *software* startup metrics. This choice was made because practitioners seldom speak of software startups. In practitioner literature, startups are typically assumed to be technology companies, or to either be engineering software or be using software to create value for their users. Thus, practitioners seem to think of software startup as a redundant construct when most startups indeed are focused on software. Rather than speaking of software startups, practitioners either simply speak of startups or focus more specifically on e.g. e-commerce startups. On the other hand, SE literature often refers to software startups specifically, and New Technology-Based Firm (NTBF)[2] is a long-standing construct used to refer to startups in business literature. We therefore chose to include documents speaking of startup metrics in general when those metrics were also applicable to software startups, and indeed most such documents not focused solely on financial metrics did discuss user and software metrics.

Finally, in addition to the practitioner literature some general-purpose software engineering metrics were adapted from extant academic literature. For example, some practitioner literature discussed monitoring operational efficiency and time spent on various tasks. We would occasionally adapt such generic, although nonetheless actionable, metrics to be more specific by employing existing research.

In this fashion, we sought to compile an extensive, although by no means comprehensive, list of metrics for software startups based primarily on practitioner literature. These results will be discussed in the following section.

4 Results: General-Purpose Software Startup Metrics

Much of the practitioner literature reviewed for this paper consisted of short “n metrics a startup must measure” type lists of five to ten metrics. As a result, there was a considerable amount of overlap. On the other hand, this points to there being some consensus among practitioners as to which metrics are particularly interesting. The most commonly cited metrics were: (1) user churn and user retention metrics, (2) user engagement metrics and metrics measuring user activity, (3) financial metrics focusing on short-term developments and cash burn, and (4) user-focused financial metrics such as User Acquisition Cost.

Churn, in this context, is used to refer to the number of users lost during a time period. The number of total users is important for monetizing any software. However, in the case of freemium software where the software itself is free and revenue is made through ads or in-software purchases, the number of *active* users becomes increasingly important. Such business models are common among software startups and the practitioner literature reflected this in relation to metrics.

In addition to closely measuring the number of users leaving, the activity of the users was regularly cited as an important focus as well. Simply measuring e.g. total users or registered users was considered insufficient. Instead, software startups were regularly urged to focus on measuring at least their Monthly Active Users (MAU) and, more importantly, Daily Active Users (DAU). Other such activity metrics suggested by practitioners were recency, that is, the number of days since the login of a user (i.e. aging / cohort analysis), as well as frequency of logins of the users. Furthermore, while measuring churn, software startups were also encouraged to measure user retention, that is, the number of users coming back to use the software as opposed to permanently leaving.

In addition to simply measuring how often the users used the software, software startups were urged to measure user engagement through various metrics. What exactly constitutes engagement changes based on each software, but in addition to activity, engagement was suggested to be measured by tracking what exactly the users do while using the software. For example, in a digital game, one indicator of user engagement could be the act of actually completing a task (a "quest") in the game as opposed to simply logging into the game, which in and of itself does not verify that a user is in fact doing anything in the game.

Finance-wise, software startups were recommended to focus primarily on user and customer-related metrics alongside more general financial metrics. User or Customer Acquisition Cost (CAC), i.e. the average cost of acquiring a new (paying) user, and User or Customer Life-Time Value (LTV) were the most commonly cited financial metrics. Past the user-focused financial metrics, conventional financial metrics such as revenue and profit margin were commonly discussed, although emphasis was placed especially on metrics indicating more short-term finances such as Month-on-Month growth and Monthly Recurring Revenue. Similarly, (Cash) Burn Rate and metrics related to it (e.g. monthly cash burn) were also commonly recommended for software startup practitioners to utilize. This ties to the fact that software startups are indeed typically lacking in resources, including capital, and are largely reliant on outside funding especially early on in their life cycles [44].

Past these most commonly cited metrics discussed so far, we uncovered a wide variety of metrics intended for software startup use. As our intention was not to study what *should* be measured but what *could* be measured, we chose to include any metrics thought to be relevant enough to be listed in the practitioner literature. To this end, the full list of metrics gathered during the literature review can be found in its entirety in the table below (Table 1), in alphabetical order. A total of 118 metrics were included in the table.

Some of the metrics listed are derivative. E.g. one could simply speak of customer churn in relation to the number of lost customers. However, some writers went into detail about churn-related metrics by discussing monthly churn, net churn and gross churn separately. In these cases, the sub-metrics were listed as well. On the other hand, some metrics were also merged together under more prevalent metrics. For example, "cancellations" [5] was considered related to user churn. Finally, for the purpose of making the table easier to read, only up to three references were included per metric given that e.g. Customer Acquisition Cost was discussed in 18 different references of this paper.

Table 1. List of Software Startup Metrics from Practitioner Literature

Metric and up to 3 Reference(s)	Description
Abandonment [12]	Transactions abandoned before completion
Acceptance Rate [12]	Avg. no. invites accepted by new users
Activation Rate [8, 13, 25]	Number of visitors or users performing a specif-

	ic action such as registering or installing
Active User Growth Rate [12]	No. new active users in a time period
Ad Inventory [12]	Total views of each ad in a time period
Ad Rates [12]	Value of each ad. inventory
Amplification Rate [25]	No. shares on social media per customer
Annual Contract Value [13, 17, 22]	Avg. annualized revenue per customer contract
Annual Recurring Revenue [13, 22, 41]	Predictable revenue annually (e.g. subscriptions)
Annual Run Rate [13]	Projected annualization of monthly recurring revenue
Avg. Revenue per User [13, 15, 25]	Avg. revenue per user over a time period
Avg. Revenue per Customer [13, 17, 25]	Avg. revenue per customer over a time period
Average Time on Hold [12]	Time user spends on hold when calling support
Billings [13]	Current quarter revenue plus deferred revenue from previous quarter
Bounce Rate [8, 40]	Percentage of visitors leaving website quickly
Breakeven Analysis [3]	Analysis to determine the point where revenue covers the costs of receiving it
Burn Rate [8, 15, 18]	Rate at which available capital is used
Campaign Contribution [12]	Added revenue from an ad campaign
Capital Raised to Date [23]	Amount of investment capital raised in total
Cash Flow Forecast [3]	Forecast of financial liquidity in a period of time
Cash on Hand [19]	Available capital
Churn Rate [1, 15, 17]	Lost users or customers over a time period
Click-Through Rate [12]	Visitors that clicked a specific website link
Committed Weekly Recurring Gross Profit [45]	Percentage increase in profits weekly committed recurring profit
Compounded Monthly Growth Rate [13]	Avg. % growth per month since inception, or another start point for measuring.
Content Creation [12]	No. visitors that interact with website content
Conversion Rate [1, 8, 17]	No. visitors that become users or customers, or no. users that become customers.
Cost of Goods Sold [23]	Cost of products or services sold (e.g. hosting)
Customer Acquisition Cost [3, 7, 8]	Average cost of acquiring a paying user.
Customer Acquisition cost to life-time value ratio [11, 30]	Customer Acquisition Cost vs. Customer Life-time Value
Customer Concentration [13, 31]	Revenue from largest customer vs. total revenue
Customer Count [39]	Total number of customers (paying users)
Customer Retention Cost [25]	Amount of spending on customer retention
Daily Active Users [9, 11, 13]	No. users who use the software daily
Daily Active Users to Monthly Active Users ratio [25]	A more detailed measure of user activity

Deferred Revenue [13]	Revenue received in advance of earning it
Development Time [18, 39]	Time it takes to implement a new feature
Direct Traffic [13]	Traffic coming in directly
Downloads or Installs [22]	Total amount of downloads or installs
E-mail Conversion Rate [34]	Number of recipients that e.g. became users
E-mail Open Rate [34]	No. mailing list members that open an email
Facebook Likes [5]	Number of likes on firm Facebook page
Fixed vs. Variable Costs [3]	A measure of total spending split by source.
Frequency of Logins [17]	Average frequency of user logins
Frequency of Visits [25]	Average frequency of visits to e.g. website
Gross (Cash) Burn [13]	Monthly expenses and any other outlays
Gross Churn Rate [13, 37]	Total users lost
Gross Margin [7, 13, 15]	Total revenue compared to cost of goods sold
Gross Profit [13, 17, 22]	Total revenue minus cost of goods sold
Innovation Metabolism [14]	Number of build-measure-learn cycles
Intent to Use [28, 34]	Data indicating that a new user is about to start using the software. E.g. imported custom data
Invitation Rate [12]	Avg. no. invites sent per existing user
Launch Rate [12]	No. downloaders that launched the software
Leads [29]	An estimate of prospective customers.
Lead-to-Customer rate [29]	Number of leads converted into customers
Life-time Value [3, 7, 8]	The average total revenue a customer generates
Likes per Post [34]	Likes per social media post
Load Time [9]	Time it takes for software to start or respond to user commands
Market Share [50]	
Market Value [50]	
Monthly Active Users [8, 9, 11]	No. users who use the software monthly
Monthly Cash Burn Rate [13, 30]	
Monthly Churn Rate [13]	Lost users or customers per in a month
Monthly Recurring Revenue [10, 11, 13]	Monthly predictable revenue (e.g. subscriptions)
Month-on-Month Growth [10, 13, 17]	Average of monthly growth rates
Net Adds [12]	Total new customers vs. cancellations
Net (Cash) Burn Rate [13]	Gross cash burn vs. revenue in a period of time
Net Churn [13]	New users gained vs. users lost
Net Promoter Score [9, 13, 17]	How likely users are to recommend product
Network Effects [13]	Effect of one user on the value experienced by other users (e.g. Metcalfe's Law)
New Visitors [17]	Number of new visitors

Number of Logins [5, 13]	Logins per user over a period of time
Number of Transactions [39]	Number of transactions made in a time period
Office Morale [5]	How motivated the team is
Operation Efficiency [15, 18]	Comparison of firm expenses by source
Organic Traffic [13]	Unpaid traffic from e.g. Google search results
Payback Time [25]	Time to recoup from an expense via revenue
Payment failures [45]	Number of failed transactions from users
Platform Risk [13]	Dependence on a specific platform or channel
Profit Margin [17, 25, 30]	Revenue minus cost divided by revenue for a product. Different ways to measure for e.g. Software-as-a-Service companies.
Prospects [12]	Number of users that might become customers
Purchases [12]	No. purchases made by a user in a time period
Recency [21]	Days since last visit of user
Referrals from current users [8, 27, 31]	How often current users refer new users
Referral rate [1]	Volume of referred users or purchases
Registered Users [17]	Total number of registered users
Repurchase Rate [23]	No. customers that made a purchase during the previous and current period of time
Retention Rate [1, 7, 8]	Percentage of users or customers still using the service after a period of time
Retention by Cohort [13]	% of original user base still using the software or conducting transactions in it
Return on Advertisement Spending [7]	Profits divided by advertisement spending
Revenue [5, 17, 22]	Total Revenue
Revenue Growth Rate [41, 43]	
Revenue Run Rate [11, 15]	
Reviews Considered Helpful [12]	Number of reviews considered helpful
Reviews Written [12]	Number of reviews written
Sell-through rate [13]	No. units sold in a time period in relation to the no. items in inventory at its beginning
Session Interval [17]	Average time between software use sessions
Session Length [17]	Length of average software use session
Social Media Reach [34]	Post reach within e.g. Twitter or Facebook
Sources of Traffic [17, 27, 31]	Source and volume of user traffic per source
Stability [9]	Frequency of crashes in software use
Time to Customer Breakeven [12, 30]	Time it takes to recoup from Customer Acquisition Cost
Time to First Purchase [12]	Avg. time users take to become customers
Top Keywords Driving Traffic to You [12]	Search terms used by visitors to find your site

Top Search Terms [12]	Both those that lead to revenue, and those that don't have any results.
Total Ad Clicks [12]	Number of advertisements clicked by visitors
Total Addressable Market [13, 17, 50]	Total hypothetical market size
Total Contract Value [13, 17, 22]	Value of one-time and recurring charges
Total Number of Customers [8, 32]	
Total Number of Users [5, 50]	Based on e.g. registered user accounts
Traffic [1, 5, 18]	Total number of website visits (non-unique)
Traffic-to-Leads [1]	Total traffic in relation to potential customers
Uptime [40]	Percentage of time software or website is available and operational
User Acquisition Rate [5, 9]	Total new non-paying users in a time period
User Demographics [5, 9]	Avg. age, gender distribution, location etc.
User Engagement [9, 17, 28]	Measured through e.g. login frequency. Definition depends on context.
Unique Visitors [11]	Unique website visitors during a time period
Viral Coefficient [11, 13, 32]	No. new customers each existing one converts

While the metrics listed above (Table 1) are applicable to most software startups, all metrics are ultimately context-specific to some extent and thus more useful for some software startups than others. Furthermore, metrics specifically targeted at smaller sub-sets of software startups can be more insightful to firms belonging to that sub-set than general-purpose business metrics for software startups. An e-commerce company will likely be focusing specifically on metrics related to their online store or platform, even though more universal software metrics such as Daily Active Users can supplement that data.

Furthermore, in terms of software engineering related metrics, practice-specific and method-specific metrics can be highly relevant to an organization. That is, if the work is not done ad hoc as it occasionally is in software startups [35]. Various agile methods and practices have their own metrics either built into the method (e.g. sprint duration in Scrum) or metrics for them have been suggested by extant research (e.g. [24]). Though such method-specific metrics can be applicable to any software startup choosing to employ a particular method, they are arguably not universally applicable to software startups. Methods and practices used to engineer software are highly diverse, with practitioners often choosing to use in-house methods created by tailoring existing methods and practices [16]. This is also the case for software startups [35]. Indeed, few method-independent SE metrics were discussed in the literature.

5 Discussion and Conclusions

In this paper, we have presented an unverified list of software startup metrics primarily based on practitioner literature (Table 1). Though we have provided an extensive list of various metrics for software startup practitioners, we have offered little verification for any of the listed metrics. The list can offer ideas for what to measure but cannot verify what effect tracking any of these metrics may have for a software startup. We can also not offer any recommendations on which metrics to use to achieve different goals. Furthermore, though the list is extensive, it is not comprehensive: many other metrics, especially more context-specific ones, can be conceived. Additionally, various conventional SE metrics and financial metrics not included in the list can likely be applied to software startups even though they were not present in the literature reviewed.

Another issue with the data is that many of the practitioner accounts dealing with software startup metrics come from the point of view of third parties. I.e. rather than being written by software startup practitioners for software startup practitioners, many of the writers are investors, startup advisors, and other external affiliates. Thus, many of the metrics discussed in the practitioner accounts reviewed for this paper were metrics (potential) investors typically wish to see when considering investing in a software startup. On the other hand, some of the practitioner accounts also discussed metrics mainly intended for internal organizational use in software startups such as operational effectiveness.

Furthermore, data and metrics are powerful tools but need to be utilized in a fitting fashion to be useful. It is important to measure relevant phenomena and to use the data to make decisions in a context-dependent fashion. More universally applicable metrics such as the ones presented in this paper can offer a useful starting point for practitioner organizations. However, more context-specific metrics such as e-commerce startup metrics can offer more valuable insights inside that context. Furthermore, every company can devise metrics unique to that company specifically that may offer even better insights into their business specifically. For example, a software startup whose main product is an online game may use metrics related to in-game data from that particular online game in order to improve the product.

Nonetheless, despite its limitations, the list of metrics presented in this paper is both a part of on-going research as well as a research proposal. Those interested in software startups and their use of metrics can make use of this list in further studies in that area. Further research on the topic could seek to study some individual metrics or groups of metrics in empirical settings, or to categorize the metrics to better suit certain contexts such as the aforementioned e-commerce domain while also adding more context-specific metrics related to that area.

On the other hand, practitioners affiliated with software startups may utilize the list to potentially gain new insights into what metrics software startups could measure. We urge any interested practitioners to view the list through the lens of their particular business and to use their own judgment on which metrics could be potentially relevant for their business. While there exists some consensus on what is important to measure in software startups in the practitioner literature reviewed for this study, we can currently offer no empirical validation in favor of any of them.

To summarize, we conducted a multi-vocal literature review primarily focused on practitioner literature. We combined an extensive list of software startup metrics (Table 1 in section 4) that software startups could measure. Based on the literature, practitioners generally recommend that software startups focusing on measuring:

- User retention and user churn
- Active users and user engagement
- Short-term focused financial metrics such as month-on-month growth and cash burn rate
- User-focused financial metrics such as User Acquisition Cost

While there was a large amount of variety in the metrics discussed in the practitioner literature, these were the most prevalent metrics among the literature reviewed. However, ultimately every business is unique and needs to establish separately which metrics are relevant for that particular business. Similarly, different metrics serve different purposes. Financial metrics may serve to indicate that something is wrong with a software but will likely not help in understanding what that might be.

References

1. Alexeeva, D. (2018). 8 Startup Metrics You Should Care About First. <https://eze.tech/blog/8-startup-metrics-you-should-care-about-first/>
2. Almus, M., and Nerlinger, E. A. (1999). Growth of New Technology-Based Firms: Which Factors Matter? Small Business Economics, 13(2), 141-154.
3. AppsterHQ. 4 Financial Metrics That All Startups Should Measure. Retrieved 07 Oct 2018 from <https://www.appsterhq.com/blog/4-financial-metrics-startups-measure/>
4. Atterer, R., Wnuk, M., and Schmidt, A. (2006). 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.
5. Belosic, J. (2018) 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>
6. Blank, S. (2013). Why the lean start-up changes everything. Harvard Business Review, 91(5), 63-72.
7. Bloem. C. 5 Performance Metrics Your Small Business Should Track. Retrieved 07 Oct 2018 from <https://www.inc.com/craig-bloem/5-key-metrics-every-early-stage-business-must-track.html>
8. Brookes, I. (2017). Startup metrics for customer traction. https://www.cakesolutions.net/companyblogs/startup-metrics-for-customer-traction.
9. Causey, A. (2018). How to Measure Your Mobile App Startup's Performance. <https://dzone.com/articles/how-to-measure-your-mobile-app-startups-performanc>
10. Cook, A. (2018). Top 5 Startup Metrics to Show Traction. <https://five23.io/blog/top-5-startup-metrics-to-show-traction/>
11. Corporate Finance Institute. Startup Valuation Metrics (for internet companies). Retrieved 07 Oct 2018 from <https://corporatefinanceinstitute.com/resources/knowledge/valuation/startup-valuation-metrics-internet/>
12. Croll, A., and Yoskovitz, B. (2013). Lean Analytics: Use Data to Build a Better Startup Faster. O'Reilly Media Inc.

13. Desjardins, J. (2017). 34 Startup Metrics for Tech Entrepreneurs. http://www.visualcapitalist.com/34-startup-metrics-founder-know/
14. Dolginow, D. (2011). Why Product Metabolism Is Every Startup's First KPI. <https://venturefizz.com/stories/boston/why-product-metabolism-every-startup-s-first-kpi>
15. Ehrenberg, D. (2014). The Seven Startup Metrics You Must Track. <https://www.forbes.com/sites/theyec/2014/06/20/the-seven-startup-metrics-you-must-track/#2760f134725e>
16. Ghanbari, H.: Investigating the causal mechanisms underlying the customization of software development methods. Uni. of Jyväskylä: Jyväskylä Studies in Computing, 258 (2017).
17. Gorski, T. (2016). 21 Most Important SaaS Startup Metrics. <http://www.saasgenius.com/blog/21-most-important-saas-startup-metrics>
18. Greenberg, O. (2016). What are the best metrics to measure funded startup growth? <https://kurve.co.uk/what-are-the-best-metrics-to-measure-funded-startup-growth/>
19. GrowthWright (2018). Financial Metrics Every Startup Should Measure. https://growthwright.com/blog/financial-metrics-every-startup-should-measure/
20. Haden, J. (2013). Best Way to Track Customer Retention. <https://www.inc.com/jeff-haden/best-way-to-calculate-customer-retention-rate.html>
21. Harley, A. (2016). Frequency & Recency of Site Visits: 2 Metrics for User Engagement. <https://www.nngroup.com/articles/frequency-recency/>
22. Jordan, J., Hariharan, A., Chen F., and Kasireddy, P. 16 Startup Metrics. https://a16z.com/2015/08/21/16-metrics/
23. Kraus, E. (2016). 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>
24. Kupiainen, E., Mäntylä M. V, and Itkonen J. (2015), Using Metrics in Agile and Lean Software Development - A Systematic Literature Review of Industrial Studies.
25. Lovelace, N. (2018). How to measure your startup's success. <https://medium.com/kandu/how-to-measure-your-startups-success-34b8aad7516b>
26. McAfee, A., Brynjolfsson, E., Davenport, T. H., Patil, D. J., & Barton, D. (2012). Big data: the management revolution. Harvard business review, 90(10), 60-68.
27. McClure, D. (2007). Product Marketing for Pirates: AARRR! (aka Startup Metrics for Internet Marketing & Product Management). http://500hats.typepad.com/500blogs/2007/06/internet-market.html
28. McNally, K., and 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/>
29. 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/>.
30. Nadel, P. (2016). 12 KPIs you must know before pitching your startup. <https://techcrunch.com/2017/02/04/12-kpis-you-must-know-before-pitching-your-startup/>.
31. Parsons, N. (2018). What Metrics to Track (and What Not to Track). <https://www.liveplan.com/blog/what-startup-metrics-should-i-track/>
32. 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/>
33. Patel, N. (2015). Measuring Retention for Startups. <http://www.neilpatel.co/measuring-retention-for-startups/.>

34. Patel, N. (2016). 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>
35. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., and Abrahamsson, P. (2014). Software development in startup companies: A systematic mapping study. Information and Software Technology, 56(10), 1200-1218.
36. Pinero, B. (2017). Data points: what should your startup measure? <https://www.intercom.com/blog/data-points-what-should-your-startup-measure/>
37. Ries, E. (2011). The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses. Random House LLC.
38. Ross, S. A. (1995). Uses, Abuses, and Alternatives to the Net-Present-Value Rule. Financial Management, 24(3), 96-102.
39. Singer, S. (2016). How To Measure Your Startup's Performance (Pt. 2). <https://magazine.startus.cc/measure-performance-startup-pt-2/>
40. StartupBahrain (2017). 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/>
41. Straubel, E. Getting Funded: Part 5 (The metrics). Retrieved 07 Oct 2018 from https://www.bigroomstudios.com/startups/startup-metrics/
42. Suster, M. (2011). How Startups Can Use Metrics to Drive Success. <https://bothsidesofthetable.com/how-startups-can-use-metrics-to-drive-success-d361b8989f5d>
43. Tyson, L. (2016). The Ultimate Startup Metrics Guide: 5 KPIs That VCs Recommend. <https://www.geckoboard.com/blog/ultimate-startup-metrics-guide-5-kpis-vcs-recommend/>
44. 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., and Yagüe, A. (2016). Software Startups – A Research Agenda. e-Informatica Software Engineering Journal, 10(1), pp. 89-123.
45. Young Entrepreneur Council (2013). 12 Success Metrics Your Startup Should be Tracking. <https://www.huffingtonpost.com/young-entrepreneur-council/12-success-metrics-your-s_b_3728052.html>
46. van Moorsel, A. (2001). 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 Company.
47. Walker, J. S. (2014) Big Data: A Revolution That Will Transform How We Live, Work, and Think. International Journal of Advertising, 33(1), pp. 181-183.
48. Wang, X., Edison, H., Bajwa, S. S., Giardino, C., and Abrahamsson P. (2016). 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.
49. Warren, P., Gaskell, C., and Boldyreff, C. (2001). Preparing the ground for Website metrics research. In Proceedings of the 3rd International Workshop on Web Site Evolution. WSE 2001.
50. Weiss, M. (2017). Top Startup Traction Metrics Considered By Seed Round Investors. <https://www.rocketspace.com/tech-startups/top-startup-traction-metrics-considered-by-seed-round-investors>

The buried presence of entrepreneurial experience-based learning in software startups

Dron Khanna and Xiaofeng Wang

Free University of Bozen-Bolzano, Bolzano, Italy
{dron.khanna,xiaofeng.wang}@unibz.it

Abstract. In less than a decade, software startups gained a lot of attention. Scholars and practitioners got the opportunity to inspect different aspects of software startups, including success, failure, challenges and growth. A startup team is a crucial part of an entrepreneurial endeavour, and learning from experience plays a vital role for both individual entrepreneurs and the team. With experiential learning, an entrepreneur requires less guidance at initial startup stages and can focus better on the objectives. Sharing of the gathered experiential learning with other team members is crucial for building a good team and making meaningful progress towards the vision of the startup. However, startup teams often neglect to reflect on their experience. Based on the gathered literature, this paper proposes a conceptual framework to understand how software startup teams could obtain entrepreneurial experience-based learning. Moreover, the paper highlights the challenges of entrepreneurial experience-based learning confronted by software startups.

Keywords: Experiential Learning · Entrepreneurial Learning · Software Startups · Entrepreneurial Experience · Reflection

1 Introduction

In less than a decade, software startups gained a lot of attention [1]. A startup's goal is to solve customers' urgent problem, and doing so they work under extreme unpredictability and encounter various challenges [2],[3],[4]. Startup teams, while working on various versions of a prototype [5] and busy at satisfying customers' need, often neglect to reflect on their experience[6]. Experience of an entrepreneur in a startup team is a vital aspect for a startup. There are various advantages of reflecting on experience; entrepreneurs can quickly grasp the challenges, coordinate better with the team members, align themselves to others in terms of fixed objectives, and require less guidance at the early stages [7], [8]. An experience that a learner [1] encounters is basically the complete response to an event. This event could be formal (planned or deliberately done [9]) or informal (unintentionally happens) [10].

In order to obtain the learning from experience, an entrepreneur should reflect on the experience. A reflection is a form of a reaction that is carried out by a

[1] In this article, we refer to an entrepreneur as a learner

learner on an experience. The resulting outcome which is learning [10] can be defined as concept of experience plus reflection [11], [12].

This type of intentional learning [13] in which adult learners [14] are attentive about what they are learning [10], what is performed on the experience [15] or in which the learner aims to preserve [16], is our focus of interest. It is important to promote this experience-based learning in the working environment [17], [18], [19]. Our target is not on the learning which is intended to adult learning in classroom settings [20].

Various authors have mentioned the challenges that occur in software startups. Bosch et al. have argued the challenge of whether it is worth to invest resource to scale the product idea of software startup [3]. Giardino et al. have provided challenges that startups have to face from forming the idea journey to the initial prototype deliver to market [2]. But there is less mentioned about the experiential learning and the challenges involved in the context of software startups. Therefore the research questions this paper addresses are:

RQ 1.) How to obtain the entrepreneurial experience-based learning inside the software startup teams?

RQ 2.) What are the challenges of entrepreneurial experience-based learning inside the software startup teams?

To answer the research questions the article provides a conceptual framework for entrepreneurial experience-based learning inside the software startups. The paper is organized as follows: Section 2 presents related work on experience-based learning. Section 3 describes the conceptual framework formalized based on the gathered literature. Section 4 states the discussion section and finally concludes the paper.

2 Related Work

The existing literature shows that various scholars mentioned the experience-based learning terminology in three ways,"learning from experience", "experiential learning" and "experience-based learning". The degree of sameness between these phrases is much higher than a degree of dissimilarity [21].Irrespective of the terminology mentioned the essence lies in the experience that is encountered by the learner. It is the main source of learning in order to draw the experience-based learning [21],[22],[23],[24]. The other similar terminology to experience-based learning is "entrepreneurial learning" because it is defined as a learning where the entrepreneur establish competence through experience [25],[26], [27].

Based on experiential learning, author Matsuo [17] describes a framework in which five facilitators, that assist the learning process based on experience. Namely,"seeking challenging task", "critical reflection", "enjoyment of work", "learning goal orientation" and "development network" are the mentioned facilitators in his framework. The utmost two facilitators are antecedents of the first three that assist the progress of experiential learning. Also, "learning goal orientation" and "development network" are set as drivers that trigger the initial three facilitators. "seeking challenging task", "critical reflection" and "enjoyment of work" directly impact on the D.A Kolb's experiential learning model [28]. Matsuo, proposed the framework to overcome the weak points (social factors, critical reflection, and the meta-learning process [29], [30], [31])of Kolb's experiential learning cycle [17].

Few authors have mentioned about challenges of experience-based learning while applied in a new environment from previous work experience. Toft-Kehler et al. state that for an individual, one of the challenges is to generalize the previously obtained experience and apply it to the new settings [32]. Gathered knowledge by an individual could vary according to previously encountered events. Generally, the newly formed firms face challenges and if the new environment setting is intricate or tricky, it is difficult to transfer the former experience. The other challenge that article describes is that the earlier obtained learning cannot be predicted as additive and self-regulating with every consecutive effort [32]. Aarstad et al. describe the challenge in regards to a novice and experienced entrepreneur. The way experiences entrepreneur respond to challenges in the firm differs from a novice. This due to the experience that is accumulated in an individual because of the daily learning activities and handling various critical events. Experience of entrepreneur helps to well-defined and lay strategies in order to deal with a critical problem in the firm, while novice entrepreneur faces difficulties to do the same [33]. Few entrepreneurs carry out a project in an improved manner if they have experienced a tough or bad period and some in worsened manner if they have experienced a good period. In general, an entrepreneur is subjected to be over-optimistic about the project. Sometimes, even too intellectual and over-confident behaviour towards the project in their initial ventures [34]. One learning challenge states the reflection of learner own and internal energy interaction with the experience. Experiential learning requires the holistic [21], [15] involvement of the learner. Sometimes a learner is unwilling to reflect upon an experience due to their understanding or assumptions on a subject [12]. The essence could exist in experience but sometimes it does not lead to learning. This is due to that fact that not all humans learn evenly from the same kind of work nominated to them [35], [36]. The table 1 below summarizes the gathered challenges from the literature:

Table 1. Challenges involved with entrepreneurial experience-based learning

No.	Challenges	Reference
1	Generalizing the old experience and applying to new venture.	[32]
2	Previously obtained experience cannot be predicted as self-additive and regulative.	[32]
3	Difference between the experience of novice or experience entrepreneur.	[33],[7]
4	Over optimistic and intellectual behavior in there initial ventures.	[34]
5	Learner unwilling to reflect upon an experience.	[12]
6	Same kind of work does not lead to equal experiential learning.	[35],[36]

3 Conceptual Framework

An entrepreneur do learn from experience [26] and in daily routine horde of experience takes place [37]. It is important that learner should be mentally aware when the events take place. Figure 1 provides the conceptual framework to cast the experience into learning. The framework comprises four elements of learner or entrepreneur, reflection, learning and experience. This framework is the evolution of the previous work done [6]. Based on the gathered literature, various changes were done in order to evolve the framework more concrete. One of the most important reasons to modify the framework was to perform reflection on experience but free from time intervals. Previously, the work was based on agile retrospectives where the obtained learning was at various intervals that is whenever the retrospective was carried out by the team. As experience takes place irrespective of the time [37] boundaries, therefore, reflection should be performed too regardless of specific interval.

Earlier the work showed the process to obtain learning from experience in a team by applying agile retrospectives. According to agile manifesto twelfth principle "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly" [38] a team should reflect on various intervals (sprints/iterations). A retrospective is one of the agile practices in order to perform reflection. Various authors have mentioned the ways to perform retrospectives. Commonly adopted by the team is the five-step process [39] by Derby et al.[40]. These steps are "1. Set the stage, 2. Gather Data, 3. Generate Insights, 4. Decide What to Do, and 5. Close the Retrospective" [39]. Three questions are discussed during the retrospective sessions: What went well? What did not go well? and What could be done? [41], [42] to make the enriched future sprints.

As reflection is the process that can be carried out anytime, hence we matured the work by modifying and removing some elements based on the gathered literature. Source and type of experience encountered by the learner are specified in details in the framework. Learning block is added and various kind of entrepreneurs are specified. Reflection done through retrospectives were removed from the framework. In addition, debriefing and reconstruction on experience are combined in the reflection block.

The experience could happen in either through an external agent or through internal emotion or sense [10]. Various authors have mentioned the types of experience present. Few to mention, the past or current experience [43],[26],[44] or the past business experience, experience from other entrepreneur [45], negative or positive experience [43]. Lamont describes the experience that an entrepreneur gather after investing time in earlier business. An entrepreneur can considerably learn from spending time after initiating or developing a company[45]. One of the essential factors which influence the performance of an entrepreneur is dependent on the earlier experience obtained by involvement in the previous business [32],[46]. The experience obtained can also scale in terms of negative or positive experience. An entrepreneur inclines to apply the positive practice again and abandon the negative ones [43].

Experiential learning could also variate in terms of the type of entrepreneurs. For example, depending on the difficulty of the task, an entrepreneur could experience differently[35]. Ucbasaran et al. classify four type of entrepreneurs habitual, portfolio, serial and novice entrepreneur [47]. These entrepreneurs can obtain the experience by practising [25] the events. This is due to the fact that when there is diversity in the environment or flow of an event, an entrepreneur can identify them. This raises the forthcoming concern and hence entrepreneur should notice it and increase there experience about the environment [25].

They occur in a learner's life by the earlier or current events or activities[21]. It is vital to recognizing those events, type of events and their meaning [48], [35]. The experience could also be obtained by the experience from others, observing other entrepreneur and their gathered information. In this way, entrepreneur lowers the own competence to learn directly. But this is useful in the case when the experience to be learned from others is available in sufficient amount in the entrepreneurial learning environment [49]. Some entrepreneur having experience with previous failures or negative experience in the business are less likely to encounter negative events in future business [50],[44],[51] and positive experience help to start a new business faster [51].

Learners reflection on the experience is a crucial component of this framework. In order to learn from experience, experience is not just enough, debriefing and reflecting on the experience is crucial too [21], [52], [54], [55], [14]. Debriefing in terms to experience is specifying the events in every detail or keeping the note for actions. It assists the reflection in formatted or structure manner, through which experience is used for learning. Debriefing is a stage which could occur either shortly after the experience was encountered or sometimes later [37]. If the learner encounters the immediate experience, commonly it is associated with deliberate or planned event [9]. In order to have a planned experience, there are three stages: preparation, engagement in the event and transforming into what has been experienced [10]. Three questions that could be done during the debriefing stage are: What happened?, How did the learner feel? What does it mean? [37]. Critical analysis and synthesis on the experience is also enumerated at the debriefing and reflection [52] stage.

The meaning of reflection here with debriefing is to specify the buried thought on experience which learner wants to take into consideration, in return to obtain learning. Reflection is deliberately viewing and thinking about the experience and then analyzing them [10], [53]. To reflect, the four styles of reflection the learner could consider are telling, writing, multimedia and activities. Telling could be done by storytelling, presentations, discussions and talking informal way with other. Writing, on the other hand, includes directed writing, a case of sheets (with a script, drawings and maps), a diary of tacking the thoughts with respect to a period of time. Multimedia incorporates videos, photos, a collage of illustrations or images or some visual representations. Finally, the activities could be accomplished by interviewing someone or spending time for some particular activity [9]. According to [9] some ways which help a learner to reflect on experience are listed in the table where a learner should spend:

– time alone and in silence environment, thinking quietly [10]
– be aware off and carefully observe the current event [10]
– examine the event [10] or task with deep understanding and the objective why it is carried out
– thinking and making sense of earlier experience [10] and connecting dots with the current experience
– should allow oneself to be emotional while reflecting on thoughts

To complete the learning process after debriefing and reflection, a learner should then evaluate and reconstruct in the light experience [21]. While reconstructing the experience, a learner can do it individually, collectively or both[21]. The main objective for this step is to make the experience ongoing and persuasive. A learner should define the reflected experience, agree and deal with the emotion and attitude. Basically to be compatible, clear and more understanding of the new experience or information, a belief which was reflected [10]. The learning element here involves three crucial elements of the learner's involvement (capability of his mind, awareness, sensitivity) [21]. Experience-based learning includes various elements. Learning occurs once the learner has pursued these elements [56]. Furthermore, a learner should continuously reflect upon the experiences encountered which help to enumerate and reconstruct them into the buried understanding [21]. To obtain and share the learning inside the team

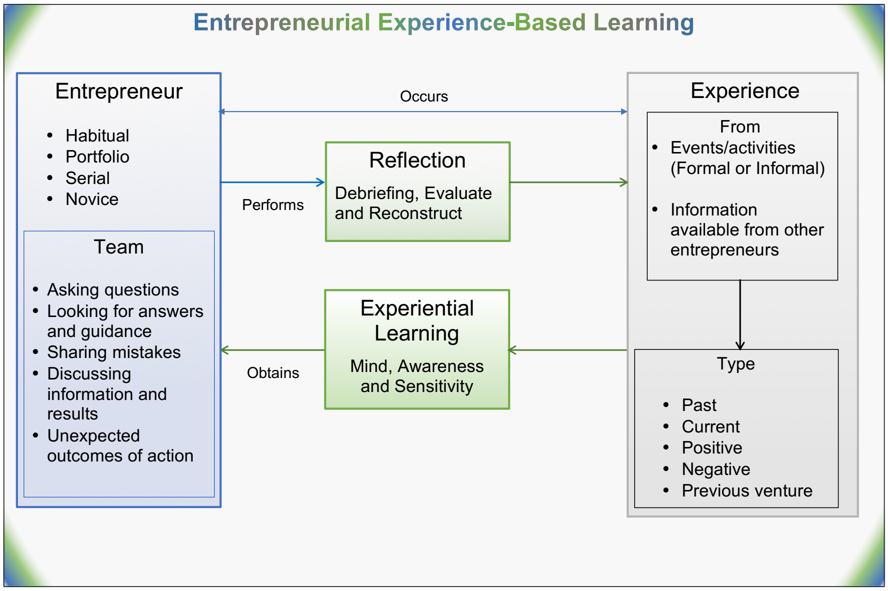


Fig. 1. Conceptual framework for entrepreneurial experience-based learning

it is important to perform the reflection with the team members. This process of reflection and action upon it could be carried out by a group of individual learners in teams by asking questions, looking for answers, sharing mistakes and unexpected outcomes and discussing them [57]. While doing so, learners share their reflected experience. At few instances, learners learn from others shared experience, but if the information available is in abundance [49].

4 Discussion and Conclusions

This study describes the entrepreneurial experience-based learning in software startups. Previous studies on software startups have neglected the importance of entrepreneurial experience-based learning inside the software startups teams. We have proposed a conceptual framework in order to obtain the entrepreneurial experience-based learning inside the software startup teams. The proposed framework is the evolution of the past work done [6] based on the gathered literature. The previous work included agile retrospectives in order to reflect on the experience. Currently the framework is free from time-dependent retrospectives. Reflection can be performed at any instance of time in order to obtain experiential learning. Also, the article suggests the tips and techniques of performing reflection on experience. The threats to validity that are worth to discuss [58]. One of the most important threat is the validity of the framework. In order to mitigate this risk, we will do the validation of the framework with several software startups. It is an ongoing work and the future literature review would contribute to the framework, by making learning from experience a continuous loop. Moreover, we will also update the literature with other experiential learning challenges in the context of software startups. Finally, we would provide an empirical tested conceptual framework for entrepreneurial experience-based learning in software startups.

References

1. Unterkalmsteiner, M., Abrahamsson, P., Wang, X., Nguyen-Duc, A., Shah, S., Bajwa, S. S., ..., Edison, H.: Software startups–a research agenda. In: e-Informatica Software Engineering Journal, 10(1).(2016)
2. Giardino, C., Bajwa, S.S., Wang, X., Abrahamsson, P.: Key challenges in early-stage software startups. In: International Conference on Agile Software Development, Springer, Cham., pp.52–63. (2015)
3. 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: Lean Enterprise Software and Systems, Springer, Heidelberg, pp.1–15. (2013)
4. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: A systematic mapping study. In: Information and Software Technology, 56(10), pp.1200–1218. (2014)
5. Khanna, D., Nguyen-Duc, A., Wang, X.: From MVPs to pivots: a hypothesis-driven journey of two software startups. In: International Conference on Software Business. (2018) arXiv preprint arXiv:1808.05630

6. Khanna, D.: Experiential Team Learning in Software Startups.In: International Conference on Agile Software Development. Springer, Cham. (2018)
7. Kim, P.H., Longest, K.C.: You can't leave your work behind: Employment experience and founding collaborations. In: Journal of Business Venturing, 29(6), pp.785–806. (2014)
8. Wasserman, N.: The founder's dilemmas: Anticipating and avoiding the pitfalls that can sink a startup. Princeton University Press. (2012)
9. Collier, P.J., Williams, D.R.: Reflection in action. CR Cress, PJ Collier, (50), pp.83–97. (2005)
10. Boud, D., Keogh, R., Walker, D.: Reflection: Turning experience into learning. Routledge. (2013)
11. Dewey, J.:Experience and Education Kappa Delta PI Lecture Series. In: Collier-Macmillan Books 1963, London, (1938)
12. Fowler, J.: Experiential learning and its facilitation. In: Nurse Education Today, 28(4), pp.427–433. (2008)
13. Tough, A.: The Adult's Learning Projects. A Fresh Approach to Theory and Practice in Adult Learning. (1979)
14. Lewis, L.H., Williams, C.J.: Experiential learning: Past and present. New directions for adult and continuing education,(62), pp.5–16. (1994)
15. Boud, D., Pascoe, J.: Conceptualizing experiential learning. D. Boud & J. Pascoe. Experiential Learning: Developments in Australian Post Secondary Education. Sydney: Australian Consortium on Experiential Education. (1978)
16. Knapper, C., Cropley, A.J.: Lifelong learning in higher education. Psychology Press. (2000)
17. Matsu, M.: A framework for facilitating experiential learning. In: Human Resource Development Review, 14(4), pp.442–461. (2015)
18. Illeris, K.: What do we actually mean by experiential learning? In: Human Resource Development Review, 6(1), pp.84–95. (2007)
19. Yamazaki, Y., Kayes, D. C.: An experiential approach to cross-cultural learning: A review and integration of competencies for successful expatriate adaptation. In: Academy of Management Learning & Education, 3(4), pp.362–379. (2004)
20. Coleman, J.S.: Differences between experiential and classroom learning. Keeton and. (1976)
21. Andresen, L., Boud, D., Cohen, R.: Experience-based learning. Understanding adult education and training, (2), pp.225–239. (2000)
22. Kolb, A.Y., Kolb, D.A.: Experiential learning theory: A dynamic, holistic approach to management learning, education and development. The SAGE handbook of management learning, education and development, pp.42–68. (2009)
23. Clark, R.W., Threeton, M.D., Ewing, J.C.: The Potential of Experiential Learning Models and Practices in Career and Technical Education and Career and Technical Teacher Education. In: Journal of Career and Technical Education, 25(2), pp.46–62. (2010)
24. LealRodríguez, A.L., Albort-Morant, G.: Promoting innovative experiential learning practices to improve academic performance: Empirical evidence from a Spanish Business School. In: Journal of Innovation & Knowledge. (2018)
25. Zheng, W., Xu, M., Chen, X., Dong, Y.: Who is shaping entrepreneurial experience? A multiple case study of Chinese entrepreneurial learning. In: Management Decision, 55(7), pp.1394–1409. (2017)
26. Sardana, D., Scott-Kemmis, D.: Who learns what? A study based on entrepreneur from biotechnology new ventures. In: Journal of Small Business Management, 48(3), pp.441–468. (2010)

27. Politis, D.: The process of entrepreneurial learning: A conceptual framework. In: Entrepreneurship theory and practice, 29(4), pp.399–424. (2010)
28. Kolb, D. A.: Experiential learning: Experience as the source of learning and development. Englewood Cliffs, NJ: Prentice Hall. (1984)
29. Holman, D., Pavlica, K., Thorpe, R.: Rethinking Kolb's theory of experiential learning: The contribution of social constructivism and activity theory. In: Management Learning, (28), pp.135–148. (1997)
30. Reynolds, M., Vince, R.: Organizing reflection: An introduction. In: M. Reynolds & R. Vince (Eds.), Organizing reflection, Aldershot: Ashgatel Gower, pp.1–14. (2004)
31. Vince, R.: Behind and beyond Kolb's learning cycle. In: Journal of Management Education, (22), pp.304–319. (1998)
32. Toft-Kehler, R., Wennberg, K., Kim, P.: Practice makes perfect: entrepreneurial experience curves and venture performance. In: Journal of Business Venturing, 29(4), pp.453–470. (2014)
33. Aarstad, J., Pettersen, I.B., Henriksen, K.E.: Entrepreneurial experience and access to critical resources: a learning perspective. In: Baltic Journal of Management, 11(1), pp.89–107. (2016)
34. Parker, S.C.: Do serial entrepreneur run successively better-performing businesses?. In: Journal of Business Venturing, 28(5), pp.652–666. (2013)
35. Matsuo, M.: The Experiential Learning Process of Japanese IT Professionals. (2006)
36. Beard, C., Wilson, J. P.: The Power of Experiential Learning. In: A Handbook for Trainers and Educators. (2002)
37. Pearson, M., Smith, D.: Debriefing in experience-based learning. Reflection: Turning experience into learning, pp.69–84. (2005)
38. Beck, K., Beedle, M., Van, B.A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J.: Manifesto for agile software development. (2001)
39. Andriyani, Y., Hoda, R., Amor, R.: Reflection in Agile Retrospectives. In: International Conference on Agile Software Development, Springer, Cham., pp.3-–19. (2017)
40. Derby, E., Larsen, D., Schwaber, K.: Agile retrospectives: Making good teams great. Pragmatic Bookshelf. (2006)
41. McHugh, O., Conboy, K., Lang,M.: Agile practices: The impact on trust in software project teams. In: IEEE Software, 29(3), pp.71-–6. (2012)
42. Ringstad,M.A., Dingsøyr, T., BredeMoe, N.: Agile process improvement: diagnosis and planning to improve teamwork. In: OConnor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, Springer, Heidelberg, (172), pp.167–178. (2011)
43. Minniti, M., Bygrave, W.: A dynamic model of entrepreneurial learning. In: Entrepreneurship Theory and Practice, 25(3), pp.5–16. (2001)
44. Rerup, C.: Learning from past experience: Footnotes on mindfulness and habitual entrepreneurship. In: Scandinavian Journal of Management, (21), pp.451–472. (2005)
45. Lamont, L. M.: What entrepreneur learn from experience. In: Journal of Small Business Management, 10(3), pp.36–41. (1972)
46. Stuart, R.W., Abetti, P.A.: Impact of entrepreneurial and management experience on early performance. In: Journal of Business Venturing, 5(3), pp.151–162. (1990)
47. Ucbasaran, D., Westhead, P., Wright, M.: Habitual Entrepreneurs, Edward Elgar Publishing, Northampton. (2006)

48. Krogstie, B.R., Divitini, M.: Shared timeline and individual experience: Supporting retrospective reflection in student software engineering teams. In: Software Engineering Education and Training, CSEET'09. 22nd Conference, pp.85–92. (2009)
49. Lévesque, M., Minniti, M., Shepherd, D.: Entrepreneurs' decisions on timing of entry: Learning from participation and from the experiences of others. In: Entrepreneurship Theory and Practice, 33(2), pp.547–570. (2009)
50. Ucbasaran, D., Westhead, P., Wright, M., Flores, M.: The nature of entrepreneurial experience, business failure and comparative optimism. In: Journal of Business Venturing, 25(6), pp.541–555. (2010)
51. Long, D., Dong, N.: The effect of experience and innovativeness of entrepreneurial opportunities on the new venture emergence in China: The moderating effect of munificence. In: Journal of Entrepreneurship in Emerging Economies, 9(1), pp.21–34. (2017)
52. Seaman, J.: Experience, reflect, critique: The end of the "learning cycles" era. In: Journal of Experiential Education, 31(1), pp.3–18. (2008)
53. Atkins, S., Murphy, K.: Reflection: a review of the literature. In: Journal of advanced nursing, 18(8), pp.1188–1192. (1993)
54. Jordi, R.: Reframing the concept of reflection: Consciousness, experiential learning, and reflective learning practices. In: Adult education quarterly, 61(2), pp.181–197. (2011)
55. Fenwick, T.: Experiential learning: A theoretical critique from five perspectives. Columbus: Ohio State University. (2001)
56. Wurdinger, S., Paxton, T.: Using multiple levels of experience to promote autonomy in adventure education students. In: Journal of Adventure Education & Outdoor Learning, 3(1), pp.41–48. (2003)
57. Edmondson, A.C.: Psychological safety and learning behavior in work teams. In: Administrative science quarterly, 44(2), pp.350-–383. (1999)
58. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. In: Empirical software engineering. 14(2), pp.131. (2009)

Business as Usual? On the Nature of Relationships in Enterprise Software Platform Ecosystems

Sabine Molenaar, Martijn van Vliet, Luc Beelen, and Slinger Jansen

Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands.
s.molenaar@uu.nl, m.vanvliet@uu.nl, l.g.n.m.beelen@students.uu.nl, slinger.jansen@uu.nl

Abstract. To obtain measurable benefit from an ecosystem, partners need to overcome the challenges they face when they join. This research aims to provide insight into whether partners are influenced by the keystone through power forms and how the partners perceive these influences. Subsequently, this research identifies possible perceived software ecosystem benefits and disadvantages that these partners experience after joining. Semi-structured interviews were conducted with both a keystone company and multiple partners within a single ecosystem. The most frequently observed power forms are reward and legitimate power. Finally, various SECO benefits and disadvantages have been identified for each partner. Thus, this research contributes to an improved understanding of partner-keystone dynamics within a single software ecosystem and provides insights beneficial to the industry.

Keywords: software ecosystems, case study, power forms, SECO benefits, disadvantages, relationships

1 Introduction

In this day and age, software developers and vendors have to consider their strategic role in a software ecosystem (SECO) to survive [15]. Organizations operating while being part of a larger SECO reap multiple benefits that they would otherwise miss out on [2, 14]. While literature regarding SECOs and their benefits are already established in current literature, explicit dynamics between the keystone and their partners are underexposed. Jansen, Brinkkemper, and Finkelstein call for case studies to be conducted in order to analyze the characteristics of individual SECOs and their effects on software vendors [11]. Moreover, Van Angeren et al. recognize the need for insight into a participant's perspective in a SECO. Questions such as what risks participants face and what benefits and drawbacks they experience within an associate model remain unanswered as of yet [1]. These dynamics between the keystone and its partners, and the benefits and drawbacks that a SECO could bring, might be more implicative for young, small, organizations. These are assumed to be more subject to change, as they are expected to rapidly adapt and evolve to become successfully established organizations.

By conducting an exploratory case study, this research provides insight into a single SECO regarding the dynamics between the keystone and its partners, mainly from a partner perspective. An advantage of using a real-world SECO as the object of study, which may not only result in industry findings as opposed to solely scientific results, is that it can help improve the industry [18]. These dynamics are analyzed by looking at multiple aspects of the relationship between the keystone and its partners. These aspects are focused on potential challenges that may arise when an organization wants to join a SECO, such as governance, as different areas of governance entail the best strategies for survival for a firm [22]. Subsequently, power forms are included to establish the origin of influences and to further illustrate the relationships [7].

By taking a closer look at the dynamics between the keystone and partners on the aforementioned aspects, the specific characteristics of the particular SECO are identified. The identification of these characteristics sheds light on possible SECO benefits or disadvantages that could be experienced. Subsequently, it grants a more complete image of the dynamics within a SECO. This contributes to the process of creating a foundation regarding the optimization of the process for future candidates that apply to join a SECO. The following main research question was formulated:

How are organizations influenced by the keystone when they join their SECO?

Subsequently, the following three sub questions have been formulated:

1. Does the keystone use power forms to influence its partners?
2. Do partners benefit from being part of the SECO?
3. Do partners experience any disadvantages from being part of the SECO?

The data required to answer these research questions was gathered through the means of semi-structured interviews.

We continue our work with a description of the literature on SECOs and power forms. Section 3 describes the research method, the data collection procedure and illustrates the SECO that was selected for this research. This section also includes our data analysis together with the organizations that participated in this research. Section 4 analyzes the results based on the research questions. The paper ends with a discussion including future research directions and a conclusion.

2 Literature Study

The literature study by Manikas was used as a starting point [17]. Further literature was added through snowballing, using both forward and backward searching.

2.1 Software Ecosystems

As stated by Jansen, a SECO is "*a set of actors functioning as a unit and interacting with a shared market for software and services, together with the relationships among them*" [15]. Partners in a SECO could be any party, on the condition that they contribute to the SECO in any meaningful and in a software related way [13]. The role of a keystone will be defined according to the description provided by Jansen, Brinkkemper and Finkelstein: "*providing a standard or platform technology that provides a fundament for (part of) the ecosystem*" [11]. Additionally, a specific type of keystone player was identified, namely the technology provider [12]. This type of keystone deploys a partnership model and desirable partners include partners that add value to the SECO, help co-innovate, help expand the SECO and promote the SECO to the "outside world" [14]. Geringer and Michael state that the culture of the partners' organization, experience, organization structure and financial position, among other factors, are taken into account by the keystone [8]. In addition, other selection criteria that could be considered include: management capabilities, established customer base, in-house knowledge (including employee skill set) and product quality [6, 23].

When the selection criteria are met, a SECO can yield benefits for partners in the SECO. For example, the involvement of partners allows organizations to focus on one specific aspect of the market, while simultaneously delegating other supportive services to smaller partners, which can result in an increase of sales for individual partners [3]. Rickmann, Wenzel and Fischbach also mention this, stating that niche players often join a SECO to gain access to customers [19]. Also, partaking in a SECO allows partners to extract value from each other, which would mean higher revenue for individual partners [14]. Barbosa and Alves state that SECOs generally decrease costs for the participants, that they support architectural decision making, allow the sharing of knowledge and that they stimulate the communication of requirements between participants [2]. These factors could also be seen as benefits from participating in a SECO and gives incentive for an organization to join an existing SECO. In order for new participating organizations to benefit optimally and thrive in a SECO, it is possible that they have to adapt to the overall climate of the entire SECO upon entry.

2.2 Power Forms

To provide more insight into partner-keystone dynamics, power forms were used to observe how they cooperate and behave in the SECO and how the keystone manages its partners [21]. French and Raven describe five forms of power in a relationship: coercive, expert, legitimate, referent and reward [7]. While French and Raven defined these bases of power in relation to a person and a social agent, Leonidou et al. [16] have adapted these to be suitable to buyer/seller relationships and define an additional form of power. In the case of the latter, these power forms are described as follows:

- **Coercive:** Threats by one firm to punish the other if it fails to cooperate and comply with its requests;
- **Expert:** The specialized and unique expertise and/or knowledge possessed by one, which is needed by the other party;
- **Legitimate:** The belief by one party that the other has a legitimate right to affect his/her behavior;
- **Referent:** The identification of one party with another, which makes one party comply with the other party's requests;
- **Reward:** The belief by one firm that another firm has the ability to mediate rewards and that it will provide these rewards if the firm complies;
- **Information:** The belief of one party that another possesses critical information, which is not available to the former.

Whenever power is exercised by one organization this can lead to compliant behavior by others [16]. While the exercised power as stated by Leonidou et al. is a work in progress and is tailored specifically to buyer/seller relationships, it is applied to keystones and partners in SECOs in this case. It is important to note, however, that such power relationships and dependencies are versatile, as they can depend on circumstance [10]. Throughout this research the power forms have been used to define the relationships and potential dependencies between the keystone and its partners, predominantly from a partner perspective.

3 Research Method

The grounded theory research method was applied, in which the data determine the theory and therefore mitigate the risk of introducing bias during the interpretation of the results. In addition, this research method allows for a wide range of different data to be used, since no particular effort was made to prove or disprove a specific hypothesis or expectation. So, the method permitted a degree of sensitivity, which allowed for the gathering of general information as well as picking up on details [20]. Ultimately, the grounded theory research method allows this research to focus on understanding the phenomena covered by the research questions, rather than explaining them [4]. In this research, the method has been applied to a case study. In the context of this research this meant that, for example, the effect the SECO has had on the partners can be observed. The data that have been gathered were used to identify the advantages and disadvantages of joining a SECO. Also, power forms utilized by the keystone within the SECO were identified. While the existence of possible power forms was known to the researchers before conducting the interviews, the effect on the grounded theory approach was mitigated by not referring to any of these power forms or their characteristics directly during the interviews. While this approach is slightly in contrast with the principles of the grounded theory method, it was required to determine the initial scope of this research. To support this approach, no hypotheses were formulated beforehand.

3.1 Data Collection

The data for this research was collected through semi-structured interviews. The questions were kept as open as possible to enable the interviewee to speak freely and add extra information when relevant. Interviews were conducted with four partners and the keystone. In case of the partners, the CEOs of the companies were interviewed. The representative of the keystone was the chief of ecosystems. The average duration of the interviews was one hour and took place at the respective organizations' headquarters. All participants received the same questions according to the interview protocol, although follow-up questions varied based on the context and the interview itself. For each interview, at least two of the researchers were present. During the partner interviews the following types of questions were asked:

– General questions (information about the organization), such as: how long has your company already been active and what is, to your knowledge, the role of your company in the SECO?
– Questions regarding the partnership with the keystone, such as the motivation to join.
– Advantages and disadvantages of being part of the SECO.
– Challenges and innovation.
– Market perspective and knowledge sharing.

Regarding the interview with the keystone, similar questions have been asked.

3.2 Data Analysis

The interviews that have been conducted have been transcribed and entered into NVIVO. Using this qualitative analysis program, potential uses of power were observed. Secondly, SECO benefits and disadvantages mentioned by partners were identified and compared. Figure 1 shows the coding nodes used in NVIVO. Some of these nodes are supported by existing literature (as described above), while others are more general. To remain open to other possibilities (outside of the literature) the "other" nodes were used to capture additional information. In case of the "sales increase" node, an increase in customers or wider reach are both included. Since no specific disadvantages were found in the literature, no examples of disadvantages were used as coding nodes. See figure 1 for an overview of the nodes.

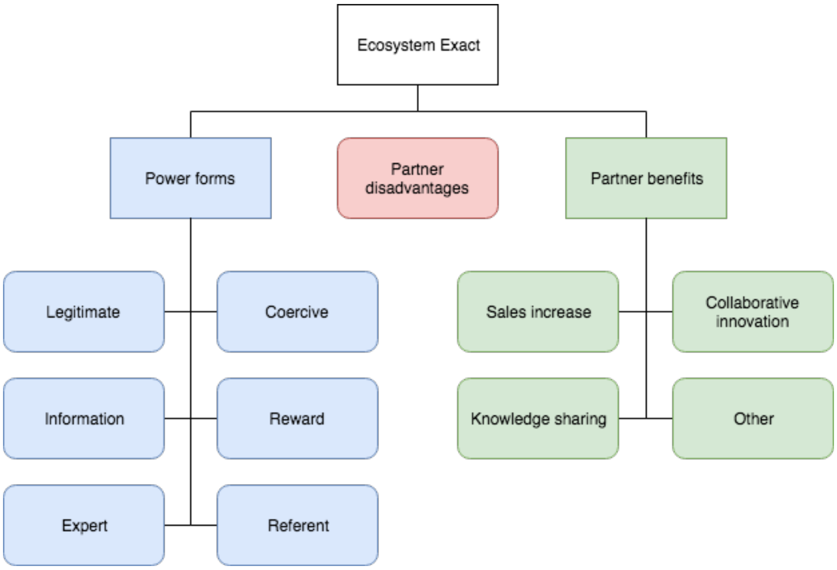


Fig. 1. Coding nodes in NVIVO.

3.3 Exact

The SECO that was investigated was that of Exact Online (simply referred to as Exact from now on). Exact has been selected, as it is the keystone of one of the biggest SECOs in the Netherlands. Exact is a Dutch software application platform provider that offers cloud business software to organizations. During the interview with a representative of Exact, the following statistics were presented (as of January 2018): they serve over 375,000 small and medium-sized enterprises worldwide and handle 2,4 billion financial transactions per month. In the Netherlands, they have 200 partners and 90,000 customers are linked. This size is important, as a product becomes more attractive when more customers use it and more suppliers provide complementary products and/or services [9]. Since the case study focuses on depth, a scope has been defined using characteristics of the SECO [17]. Firstly, the SECO boundary can be defined by the organization, more specifically, Exact Online's app center [15]. Secondly, Exact currently deploys a membership and partnership model, since some partners are required to pay a fee in order to be part of the SECO, while others have entered free of charge. Finally, the accessibility of the SECO of Exact can be described as screened, while partners are free to contribute software (such as online plug-ins), contributions need to be approved by the keystone [14].

Candidate participants that optimally suited this research adhered to the following prerequisites: participants recently joined the Exact SECO, so that the joining of the SECO was fresh in their mind and they could provide an illustration of what joining the SECO is like now, as opposed to some years ago. Secondly, they themselves, do not fulfill the role of keystone in a different SECO (as of yet). Furthermore, only small enterprises in terms of number of employees have been included. Finally, the partners in the SECO have been selected based on recommendations by Exact, to ensure that they provide an accurate representation of the SECO. Organizations of similar size, in case of the partners, have been selected, to ensure they have matching perspectives on joining a SECO and allowing for comparison of their experiences. The four partners that have been selected are part of a larger set of partners that are representatives of the criteria mentioned previously. Table 1 provides an overview of general information of the organizations that participated in this research. For one participant the information shown in table 1 and the quotes used in the results section were anonymized.

Organization	Exact	TriFact365	Invantive	vPlan	Partner Z
Year founded	1984	2011	1992	2016	2013
Product/service	Accountancy	Invoice processing	Accountancy	Planning	Data Analysis
Number of employees	1400	6	10	7	5
Year of entrance	2012 (founded)	2013	2015	2017	2014
SECO role	Keystone	Niche	Bridge	Niche	Niche

Table 1. Overview of general information of Exact and its partners.

4 Results

The findings presented in the upcoming section contribute towards the building of a theory that answers the main research question [5]. Direct quotes from the interviews were used to support all findings. These quotes, provided by the representatives of each organization, have been interpreted as the voice of their entire organization. Quotes have been translated from Dutch to English. An overview of the findings is presented in table 2 at the end of this section.

4.1 Identified Power Forms

Figure 2 depicts the power forms that could be observed within the SECO. The direction of the arrow shows where the influence is coming from (bidirectional arrows are also possible) and the color represents either a positive or negative connotation (green and red respectively), according to the party that is influenced.

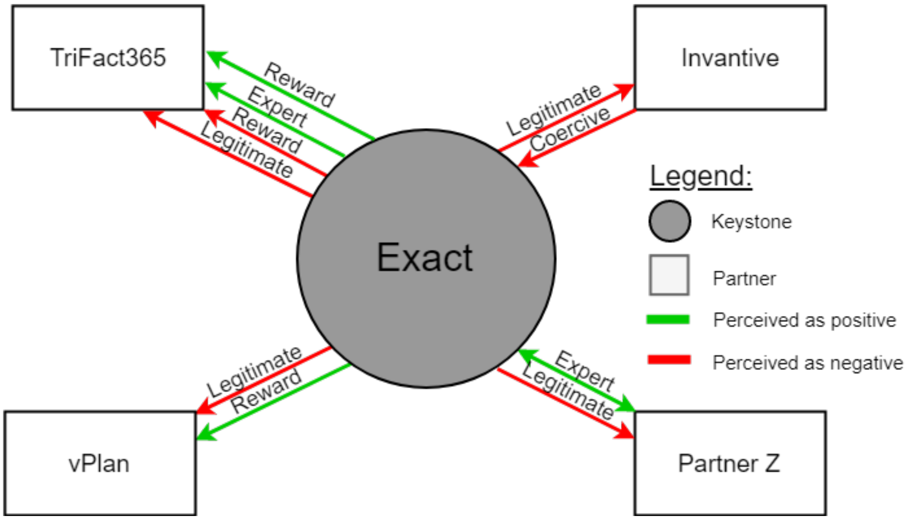


Fig. 2. Overview of the observed power forms in the Exact SECO, showing that reward and legitimate power were most frequently observed.

The most frequently observed power forms were legitimate and reward power. However, in the case of the former, the keystone met resistance even though partners felt that the keystone had a right to influence. While the partners mostly complied, the conflicts that arose negatively influenced the relationship between partner and keystone, for example a decrease in satisfaction on the partner's side as discussed by Leonidou et al. [16]. One change that the partners mentioned is that the keystone started to charge fees for being part of the SECO, while being given insufficient notice beforehand. According to the keystone, the reasoning behind the fees was that they incur costs in order to keep both the SECO and, specifically, the app center running. TriFact365 confirms this and states: "*The motivation is that they make costs for the app center and that they want compensation in return to be able to continue to innovate.*" Based on this, it seems to be a use of legitimate power. However, TriFact365 then continues saying that "*They indicated this could be discussed, we talked a little, but in the end it became clear that negotiation was not possible*". Which can be interpreted as coercive power, although no threats or consequences were expressed, whilst usually punishments are stated [16]. vPlan, however, stated that Exact was more open for negotiations: "*...we were able to make a deal in the beginning, they actually were quite approachable especially compared to others.*" Most partners deemed the current pricing dynamic to be fine, as long as they asked a fair price. Invantive, however, exerted its own coercive power in return by threatening to leave the SECO, clearly indicating a punishment for non-compliance [16]: "*At a certain point in time we said we will put it on hold, for half a year it was put on hold. We do not do [keystone's product] anymore, done.*" So, while these fees are acceptable at this point in time, an increase would likely not be tolerated or at least be deemed controversial by smaller partners. While the actual prices asked by Exact for each partner remains unclear and partner specific, the different perspectives and the highly varying reactions highlight that this is at least a sensitive subject. In the near future, Exact is planning on charging partners based on their API calls. This could lead to various consequences for the partners, the most important being additional costs, which can lead to problems. If Exact does not communicate this properly and does not notify the partners of this change ahead of time, reactions such as the one of Invantive could be expected. Moreover, this could affect different partners in different ways. While some rely on a few API calls to run their business, others need a plethora of calls to keep their product running and to meet customer demands. Another example of the use of legitimate power was that Exact asked vPlan to lower their price, otherwise the two products would cost nearly the same amount, which was not considered marketable. Notably, vPlan complied because they expected this to yield more profits than costs. Due to this increased profit, the price change was not seen as a disadvantage.

Furthermore, Exact has a vision for the future in respect to reward structures for organizations regarding the sharing of data, as these organizations help to improve the data. TriFact365 coined the idea that Exact should include successful partners in their proposition and that they, in return, are willing to share part of their revenue. During hackathons Exact also puts organizations in the spotlight, if they provided useful contributions during the event. Likewise, Exact also awards prizes via the app center, such as the "app of the month" prize, which vPlan received. According to Leonidou et al., reward power can be used to improve collaboration and productivity [16]. The aforementioned hackathons can be seen as an example of such use. Finally, TriFact365 explained that Exact imposes requirements, such as security procedures, and that they screen APIs before allowing them in the app center, which can be seen as an example of expert power. One partner also seems to exert expert power, that is to say that Partner Z aids Exact in using their data well. All in all, Exact does not seem to apply all of the aforementioned power forms. Instead, the participants mentioned reciprocity as a more prevalent factor, they are willing to share their knowledge or offer a helping hand if they can expect the same in return. TriFact365 calls it "*give and take*" and stated that both parties ask how the other is doing and provide insight on a noncommittal basis. Invantive confirmed this by saying that "*We mainly require market information and Exact requires technical information, that works well*". Additionally, vPlan also stated that their dynamic is a two way street: "*...just because it comes from both sides. Exact needs us because we provide an essential part.*"

When looking at the aforementioned power forms, especially coercive and reward, and their influence on the keystone-partner relationship, the following can be deducted: when the keystone applies its coercive power, this could presumably lead to a relationship based on "fear", as described by Leonidou et al. [16]. When this is the case, this can spread to the other partners and candidate partners, who might refrain from joining the SECO, which in turn will have a negative influence on the growth and well-being of the SECO. On the other hand, reward power, the actualization and not just the promise, has the potential to increase the attraction of partners to the keystone, as this may increase the satisfaction of partners [16]. This will benefit their relationship and have a positive influence on the SECO, because the partner is likely to share with their partners or other partners that they have a positive experience. This, in turn, can reach candidate partners that can be persuaded to join the SECO when hearing about the positive relationship between partner and keystone, as was also identified by Jansen, Cusumano and Brinkkemper [14].

4.2 Perceived Benefits of the SECO

Figure 3, similar to figure 2, shows an overview of the SECO benefits and disadvantages as stated by the partners during the interviews.

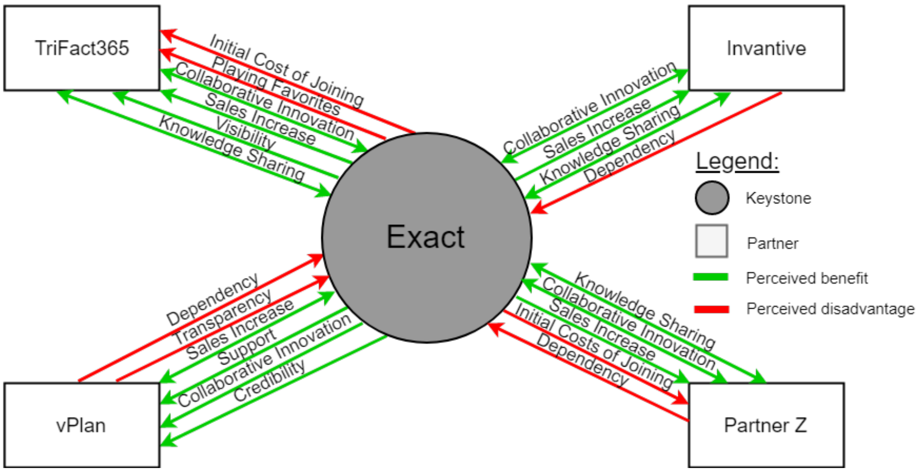


Fig. 3. Overview of the observed partner SECO benefits and disadvantages, showing that the perceived effects of partner-keystone dynamics are mostly beneficial for either party.

The three main advantages of being part of a SECO that were identified beforehand in the literature study were also encountered in the case study. One of which was an increase in sales as mentioned by Ceccagnoli et al. [3], however, not all participants mentioned they had experienced an actual increase. Invantive shared that some customers they serve via the keystone actually cost them money when looking at profit per customer: " *We also have contracts that cause a loss.*" On the other hand, all participants confirmed that, thanks to the SECO, they have a wider reach and have had the possibility to gain new customers, as was stated by Rickmann, Wenzel and Fischbach [19]. Even when partners did not personally encounter the effects of this SECO benefit (yet), they still acknowledged the potential as an important SECO benefit of being a partner.

Secondly, all participants mentioned that they share knowledge with the keystone and that they receive valuable and useful information in return, which confirms the findings of Barbosa and Alves [2]. The technology-focused organization Invantive especially appreciated the marketing knowledge the keystone possesses. When asked whether Exact is a useful source of knowledge their answer was "*it is for knowledge of the market.*" The keystone itself also mentioned that they are aware of this SECO benefit for their partners. Nevertheless, when it comes to sharing knowledge such as source code the partners remain cautious. Invantive stated that Exact "*can look at it if they want to, but cannot take it with them.*" In the case of TriFact365 the code was not at all available to Exact. However, Exact stated that they do not expect the same degree of openness from their partners. In the continuation of knowledge sharing, every participant said that they actively collaborate with the keystone in terms of innovation and that hackathons are organized frequently. This co-innovation is in line with the observations provided by Jansen, Cusumano and Brinkkemper [14]. The topics of innovations mainly include integrated services, new technologies and the renewal of APIs.

In addition to the SECO benefits that were expected due to findings in the literature study, two additional advantages could be identified during this research. Firstly, the participants mentioned that their credibility improved thanks to being part of the SECO. Customers see the keystone as a trustworthy organization and, by association, the partner organization as well. This was acknowledged by vPlan: "*What you do have, is the logo of Exact on your website which apparently means something, as it offers something recognizable for organizations.*" For smaller, relatively young organizations it felt as a big advantage to be able to express the fact that they were backed by a bigger organization that is known to public. In other words, the reputation of Exact resulted in a higher level of trustworthiness that reflects on the associated partners. Partners mentioned that this trustworthiness comes from the fact that Exact acts as a guardian regarding product quality, which guarantees a certain degree of quality of members in the SECO. This was specifically acknowledged by TriFact365: "*...they all have to score a 9 or a 10 in order to collaborate, otherwise you have to say goodbye to them (...) quality is the most important to Exact...*" Subsequently, partners mentioned that being part of the SECO can lead to increased visibility, for example due to the partner organization being named on the website of the keystone. In other words, partners acknowledged that they benefit from the marketing channel of the keystone, which by definition is larger than the individual marketing channel of a small partner. Respectively, in certain cases partners get an even bigger spotlight when Exact sees a certain potential in their product. vPlan was a great example of this phenomenon as they stated that "*As app of the month, you will get presented on the website of the app center. Potential clients see this and are easily able to make a deal at the same time with us and Exact. For us this results in a way bigger reach.*"

4.3 Perceived Disadvantages of the SECO

In addition to the identified SECO benefits, possible disadvantages also became apparent during this research. These disadvantages do not necessarily affect the individual organization specifically, but can also include aspects which the partners think should be improved. An example of such an aspect was the preferred partnerships as mentioned by TriFact365: "*I do not think the ecosystem is completely transparent.*" They think the app center should be open and that "*they should not have contracts with third parties that could cause conflicts.*" Essentially, a level playing field should be created. According to them, and Exact, they have a tendency to favor partners that have been a part of the SECO for a long time or even from the beginning. This so-called "playing favorites" is not appreciated by "regular" partners but, surely, those favorites enjoy this special treatment. In this context, the preferred partners' products are sold alongside Exact Online and, unlike the other partners, they are not required to pay a fee to be included in the app center. Also, they are more often pushed and recommended to customers than other products. Additionally, the previously mentioned spotlight of "app of the month" could also be observed as drawing away the focus from other partners. These two factors combined could potentially result in drawing attention away from other, new partners that could be of importance for innovation, growth or general longevity of the SECO.

Secondly, joining the SECO compelled the new partners to spend resources, such as time, effort and money, into linking and developing their product. After obtaining partner status, these costs also included continued management and support. This is not in line with the general decrease in costs Barbosa and Alves identified [2]. However, this also included the initial cost of joining. In addition, partners mentioned that partner managers can be of influence. Invantive said that their situation improved after they received a different partner manager: "*whom at least had more experience with complex cases.*" The same organization also stated that they think Exact has too few partners abroad. They were also required to accept medium to small organizations as customers, which they found hard to get used to saying that this period "*was a difficult time.*" Moreover, partners were concerned with what decisions Exact might make or not make, (these differed from partner to partner) and how these changes would impact them. These changes also included the transparency of the SECO. This mainly manifested itself in the fact that partners mentioned that they receive short notice, or none at all, when changes to policies (among others) are made. A specific example of this was Exact's decision to start charging fees for being included in the app center. For smaller organizations, the dependency on Exact was acknowledged to be either a disadvantage, or at the very least a risk. Organizations that have relatively little sales for other sources, such as Partner Z have little alternatives: "*...there is not really an alternative. If I want to approach Exact customers (on my own), I have to pull my app from the app center. Theoretically, I could do that, which would mean no dependency, but no business as well.*" Invantive also stated that if Exact goes bankrupt, is obtained by a third party or changes their terms and conditions "*they will have a problem* *as well.*" Thus, organizations that do not have enough customers by themselves or are not part of another SECO, share the fate of Exact and its SECO. In table 2, the main findings of this research are presented, they are relevant for both the keystone, partners as well as potential partners who want to join a SECO.

Power forms
Most frequently observed power forms are legitimate and coercive
The most striking form of legitimate power is that the keystone is planning on charging partners for the use of their API
The most stated expression of coercive power is that the keystone enforces fees for being part of the SECO
The actualization of reward power has the power to increase the attractiveness of the SECO
The principle of reciprocity is applied within the SECO, partners are willing to share their knowledge or a helping hand if they can expect the same in return
When the keystone places partners in the spotlight, this can result in exposure for partners
The keystone has a vision for the future regarding reward structures so that partners benefit from sharing with the keystone
When the keystone enforces coercive power this can lead to a relationship of fear
Perceived benefits of SECO
Increase in sales and visibility
Wider reach and possibility to gain new customers
Access to marketing knowledge of the keystone/benefit from marketing channel of keystone
The credibility and trustworthiness of partners improved due to being part of the SECO
Sharing knowledge with the keystone lead to partners receiving valuable information in return
Opportunity for collaborative innovation
Perceived disadvantages of SECO
Non-equal playing field/playing favorites
Dependency on the keystone
Lack of transparency and communication

Table 2. Main findings of this research.

5 Discussion

Due to the exploratory nature of this research, it can be used to identify research directions and provide better insight into partner and keystone dynamics. However, since the research consisted of a case study, it is difficult to generalize the results and more research into this SECO and others is desired. On the other hand, this research and its methods can easily be applied or adapted to look into other SECOs. In addition, since the characteristics of the SECO and its boundary were identified, this research can be conducted using similar scopes to confirm the findings [17]. Alternatively, it can be used to analyze the partner-keystone dynamics of this SECO, using different partners. Therefore, the research is quite scalable and can be used in other contexts. Finally, one of the strengths of this research is its use of a real-world SECO and, with that, useful insight for the industry [18].

The research was also limited by the fact that only a small subset of Exact's partners participated, while more partners would have been preferable. This was due to the limitations introduced by a narrow time window for conducting the research. More qualitative research should be performed to be able to truly confirm the findings mentioned in this research. Another limitation is the fact that all the partners that participated were suggested by the keystone of the SECO. This could have influenced the results, because the selected partners may have been more positive than without the participation of the keystone, although not all participants were equally enthusiastic about the SECO. However, this can also been seen as a advantage, because the partners have been selected in such a way that they properly represent the SECO in its entirety. Additionally, since the keystone requested them to participate they were willing to invest time and effort into the research.

As already mentioned this research can be seen as an exploratory case study. Based on the aforementioned findings we can extract the following hypotheses that can serve as a basis for our and other future research:

1. A non-equal playing field can result in potential partners not wanting to join the SECO and losing partners that are of importance.
2. Applying reward power by the keystone will increase the attractiveness of the SECO, resulting in growth and new partners.
3. Applying coercive power by the keystone will decrease the attractiveness of the SECO since it leads to a relationship of fear, scaring potential new partners away.
4. For partners, joining a SECO will result in an increase in credibility and visibility.

This research has focused on the relationship between partners and keystone, seen from the partner's perspective. However, other relationships do exist within a SECO, for example, between the keystone and the technology providers. Future research should be conducted to analyze if other relationships exist, how these can be described and what the effect is of these relationships on the SECO and keystone. Moreover, additional research aimed at discovering the intensity of the identified power forms in this study could also lead to further insights.

6 Conclusion

The main objective of this research was to gain a better understanding of the influence of the interactions between organizations and the keystone of a SECO. This research provides multiple insights into partner and keystone dynamics within an SECO, its main contribution to the field of SECO being the partner perspective. More specifically, the conducted interviews illustrated how small organizations are influenced by the keystone when they join their SECO. Three main factors have been discussed: power forms, SECO benefits and disadvantages. Based on the results presented in the previous section it became apparent that partners are required to ensure their product(s) meet the keystone's requirements and are required to pay a fee in order to be part of the SECO. However, chances are that the partners are rewarded for their contributions to the SECO. Secondly, the SECO benefits as stated by the literature have been confirmed by the partners. In addition, two new advantages were identified: visibility and credibility. Finally, disadvantages could be observed as well, although these could more accurately be called risks and can differ per partner. All in all, after joining the SECO three of the factors described affected the organizations.

This research encompassed four partners that were active in the Exact SECO. In the future, however, more partners can be included with different backgrounds, to not only better illustrate the partner-keystone dynamics of the SECO, but also to confirm the findings presented in this research. In line with this, the research could also be applied to other SECOs inside or outside the Netherlands. Besides the new insights, improved understanding and future research directions, this research can also be of use to the organizations that participated. The keystone can use the observations and remarks made by its partners to try to improve the SECO. It also provides knowledge on what the partners struggle with or what they would like to see or do differently. The SECO benefits that were observed can be used as a means to persuade organizations to join the SECO. Finally, the organizations that are considering to join a SECO, or this one specifically, will know what to expect, what challenges they will need to overcome and what risks they are taking.

Acknowledgements: We would like to thank the representatives of Exact, Invantive, Trifact365, vPlan and Partner Z for their time and contributions to this research, as well as anonymous reviewers for their valuable feedback.

References

1. J. Van Angeren, J. Kabbedijk, K.M. Popp, and S. Jansen. *Managing software ecosystems through partnering.* Edward Elgar Publishing Ltd., 2013.
2. O. Barbosa and C. Alves. A systematic mapping study on software ecosystems. *In Proceedings of the Workshop on Software Ecosystems*, pages 15–26, 2011.
3. M. Ceccagnoli, C. Forman, P. Huang, and D.J. Wu. Co-creation of value in a platform ecosystem: The case of enterprise software. *MIS Quarterly*, 36(1):263–290, 2011.
4. Kathy Charmaz. *Constructing grounded theory: A practical guide through qualitative analysis.* Sage, 2006.
5. Juliet M Corbin and Anselm Strauss. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative sociology*, 13(1):3–21, 1990.
6. T.K. Das and I.Y. He. Entrepreneurial firms in search of established partners: review and recommendations. *International Journal of Entrepreneurial Behavior & Research*, 12(3):114–143, 2006.
7. J.R. French and B. Raven. The bases of social power. *Studies in social power*, pages 152–164, 2004.
8. J.M. Geringer and J. Michael. Joint venture partner selection: Strategies for developed countries. *Journal of International Business Studies*, 20(3):569–571, 1989.
9. E.D. Hartigh, W. Visscher, M. Tol, and A.J. Salas. Measuring the health of a business ecosystem. *In Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*, 2013.
10. J.C. Isaac. Beyond the three faces of power: A realist critique. *Polity*, 20(1):4–31, 1987.
11. S. Jansen, S. Brinkkemper, and A. Finkelstein. Business network management as a survival strategy: A tale of two software ecosystems. *In Proceedings of the first International Workshop on Software Ecosystems*, pages 34–48, 2009.
12. S. Jansen, S. Brinkkemper, J. Souer, and L. Luinenburg. Shades of gray: Opening up a software producing organization with the open software enterprise model. *Journal of Systems and Software*, 85(7):1495–1510, 2012.
13. S. Jansen and M.A. Cusumano. Defining software ecosystems: a survey of software platforms and business network governance. *Software ecosystems: analyzing and managing business networks in the software industry*, 13, 2013.
14. S. Jansen, M.A. Cusumano, and S. Brinkkemper. *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry.* Edward Elgar Publishing Ltd., 2013.
15. S. Jansen, A. Finkelstein, and S. Brinkkemper. A sense of community: A research agenda for software ecosystems. *In Software Engineering-Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on*, pages IEEE, 187–190, 2009.
16. L.C. Leonidou, B. Aykol, V. Lindsay, C.S. Katsikeas, and M. Talias. Drivers and outcomes of exercised power in buyer-seller relationships: A meta-analysis. *In 30th Annual IMP Conference*, 2014.
17. K. Manikas. Revisiting software ecosystems research: A longitudinal literature study. *Journal of Systems and Software*, 117:84–103, 2016.
18. K. Manikas and K.M. Hansen. Software ecosystemsa systematic literature review. *Journal of Systems and Software*, 86(5):1294–1306, 2013.
19. T. Rickmann, S. Wenzel, and K. Fischbach. Software ecosystem orchestration: the perspective of complementors. *Twentieth Americas Conference on Information Systems*, 2014.

20. H. Sharp, Y. Rogers, and J. Preece. *Interaction design: beyond human-computer interaction*. John Wiley & Sons, 4th edition, 2015.
21. G. Valença, C. Alves, and S. Jansen. A power perspective on software ecosystem partnerships. *In International Conference on Product-Focused Software Process Improvement*, pages Springer, Cham, 69–85, 2016.
22. C. Werner and S. Jansen. A systematic mapping study on software ecosystems from a three-dimensional perspective. *Software Ecosystems: analyzing and managing business networks in the Software Industry*, pages 59–81, 2013.
23. Y.X. Zhong and H. Ren. Partner enterprises selection for innovation alliances: A reviews perspective. *International Journal of Research Studies in Science, Engineering and Technology*, 2(10):8–16, 2015.

Platform Ecosystems for the Industrial Internet of Things – a Software Intensive Business Perspective

Dimitri Petrik[1] and Georg Herzwurm[1]

[1] University of Stuttgart, Graduate School of Excellence advanced Manufacturing Engineering (GSaME), Nobelstr. 12, 70569 Stuttgart, Germany
{dimitri.petrik, georg.herzwurm}@gsame.uni-stuttgart.de

Abstract. The global competition requires the machine tool industry to provide more flexibility and productivity to its manufacturing customers, enabled through software-intensive services. A platform approach receives an increasing attention within the machine tool industry, offering a solution to provide such services. Software platforms, adapted to the needs of the industry and used in the industrial application are also known as industrial internet of things (iIoT) platforms. Despite the growing interest among manufacturing companies in iIoT platforms, they have been limitedly researched from the economic perspective. Consequently, a further in-depth analysis of platform-based business models in the area of iIoT is still needed. Firstly, this paper offers new insights on technical and economical criteria for business models and design of existing iIoT platforms and transforms them into a taxonomy. These merged criteria provide a detailed perspective on iIoT platforms and support machine tool companies in their decision process of suitable iIoT platforms. The criteria are based on the results of 17 qualitative interviews with companies from the machine tool industry. Secondly, the identified criteria are summed up in a morphological box, in order to reduce the selection complexity of an iIoT platform by the machine tool companies and sharpen the software-intensive business models of the platform providers.

Keywords: Industrial IoT, IoT Platform, IoT Ecosystem, Business Model Analysis, Morphological Box.

1 Introduction

The machine tools industry nowadays experiences an increasing competitive pressure due to the globalization and the individualization in manufacturing, requiring more efficient manufacturing processes [1]. The German Mechanical Engineering Industry Association (VDMA) together with McKinsey have surveyed the machine tool companies and identified the customer demand for customized systems and solutions as the most relevant trend. Another finding was the increasing importance of the after-sales, ranking it as the third most relevant trend in the market [2]. Digital services are provided remotely and modularly during the whole life cycle of a machine tool, creating a steady revenue source in after-sales for a machine tool company [3-4]. The platform approach enables the provision of digital services for a variety of customers, fulfilling the flexibility needs and even building new software-intensive business models. The services are provided through enterprise applications, which are developed on specific software platforms [5], classified as iIoT platforms. Such a platform interacts with smart connected machine tools and its components across companies' borders, processing the data it receives from the machine tool. Based on the processed data the platform triggers microservices, changing the parameters of the machine tool through the data feedback loops. Accordingly, platforms play coordinating roles for connected machine tools, acting as a digital infrastructure [6-7]. Gawer and Cusumano coined the understanding of open technological platforms [8]. iIoT platforms also act as multi-sided markets [9], as machine tool companies provide applications, based on the platform, for the machine operating companies in different industries.

The current state shows, that a successful platform initialization in a machine tool industry remains a high complexity challenge for both: the platform providers and the machine tool companies, acting as a collaborating customer for a platform provider in an iIoT ecosystem [10]. The complexity is partially caused by the variety of the specific functional characteristics offered by each platform provider, by the iIoT platform evolving the machine tool company into an ecosystem and by the variety of the market-available platform solutions [11-13]. In addition, various iIoT platform providers describe only a fuzzy value proposition, without meeting the specific customer needs of the machine tool industry, as mentioned by Herzwurm [14]. However, a selection process for a suitable iIoT platform is a major challenge [15] and highly interdisciplinary, as it is crucial for the product servitization and affects stakeholders from multiple departments throughout the whole company [7, 16]. Despite the recognized potential of iIoT, machine tool companies experience difficulties to identify which iIoT platform best suits their own requirements and the current state of market hinders the formation of a "platform leader" in the machine tool industry. The current state indicates an industrial problem setting, revealing the lack on relevant technical and economic criteria for the choice of iIoT platforms from the perspective of a machine tool company as a collaborative customer. This paper is based upon the assumption, that the fragmented market for iIoT platforms (offering up to 450 market-ready solutions) causes problems for the manufacturing companies to choose the right platform. On the other hand, the practical relevance of the problem is present, as new studies conducted by the VDMA, show an increase of interest in iIoT platforms by machine tool companies. Although more than 60% of surveyed companies indicated iIoT platforms as an unknown topic or irrelevant in 2016, for 75% of surveyed companies iIoT platforms are important in 2018 [17].

Considering the current state of research on platforms, Gawer has already bridged economic and technical perspectives on platforms and offered a platform classification. However, this classification is not specific to iIoT platforms. In addition, the scientific papers about concrete design or business model patterns within the industrial application of platforms and platforms specifically used for the machine tools industry (see Chap 2.1) are still rare. As stated by Kude in the Dagstuhl position statement, the existing literature on the iIoT has mainly focused on the technical implementation and the platform literature has been mainly too generic [18]. This indicates a research gap on relevant business model criteria of iIoT platforms, which if known, collaborative customers in the iIoT ecosystems could use for interdisciplinary platform selection decisions. Hence, the overall goal is to provide relevant criteria for this selection process through a more in-depth analysis of design and business models of market-ready iIoT platforms for the industry of smart connected machine tools. Hence, this article answers the following two research questions:

- RQ1: What are the relevant criteria in the selection process of iIoT platforms by manufacturing companies for data-driven maintenance services?
- RQ2: Which market-ready iIoT platforms fulfill the identified criteria?

The structure of this paper consists of three parts. The second section of the article presents conceptual foundations and current state of research on iIoT platforms. The third section addresses both research questions, presenting at first the identified technical and economic design criteria of iIoT platforms. Criteria are based upon a multiple case study analysis of qualitative data, collected in interviews with machine tool companies. Afterwards, each elaborated criterion is applied on the market-ready iIoT platforms, in order to ensure the transferability of the identified criteria to the current state of the market for iIoT platforms. The final part presents the future research outlook and limitations.

The main result is a characteristics taxonomy for iIoT platforms, both technical and economical, integrated in Zwicky's morphological box. The morphological box could act as a decision support tool for the cross-department collaboration during the iIoT platform selection, building the main artefact of the paper. Morphological analysis as a method has been already used to gain a holistic understanding of business model concepts within a certain context [19-20]. Researchers and practitioners from the platform provider perspective could use the taxonomy for a further business model analysis of iIoT platforms, in order to better understand currently existing or even build new configurations and develop new business model patterns [21] for iIoT platforms. Practitioners from the machine tool industry could use the results in a selection process of a suitable iIoT platform. Moreover, the results can support the iIoT platform providing companies in a more precise communication of their platform design to the collaborative customers or complementors. As a result, this could increase the transparency on the design and the business models of the offered platforms, therefore involving additional collaborators in the platform-based iIoT ecosystems and stimulating the network effects [9]. Taking the research context of previously mentioned business models into account, this paper provides integrable criteria for the business model dimensions of the St. Gallen Business Model Navigator [21].

2 Industrial Internet of things and prior work

Following paragraph describes the theoretical background in the area of iIoT platforms. IoT integrates information and communications technology (ICT) with objects, connecting them with wireless and wired technologies and extending them by real-time analytics. iIoT integrates these technologies in the industrial area of application

[22]. The relation to the concept of Industry 4.0 is close, which means iIoT can be understood as the vertical and horizontal connection of people, machines, objects and ICT systems, which are real-time capable and intelligent, for dynamic management of complex systems [23]. Hence, connected machine tools act as cyber-physical systems (CPS) [24] and this transition could greatly increase the productivity and the flexibility. It is estimated, that it is possible to increase the productivity and the lifespan of machine tools up to 5%, to lower the maintenance costs between 10 and 40% and reduce the energy consumption up to 20%, if the machine tools are connected and monitored [25]. The listed benefits could be achieved through processing and analysis of machine-generated data. An intelligent machine tool could stream data considering its condition and its energy consumption, the current process or the quality of the workpiece and combine them with a cross-domain analytics. Lastly, the processing of the data appears in scalable iIoT platforms [26]. Moreover, if an iIoT platform provides open interfaces, the information could be enriched with external information sources and enable integration of third-party companies, [7, 9] thus enabling ecosystems in the area of iIoT. Compared to the customer branches, iIoT ecosystems are significantly smaller, have different requirements for platforms [27] and possess more complex structure of collaborating complementors, compared with traditional software ecosystems [28].

This paragraph shows the current state of research on the business models for iIoT and platforms. Gawer has created a unified view on open digital platforms and classified supply-chain and industry platforms as open [9]. This classification framework was only applied in the area of industrial robotics, to extract business model patterns and its dependency from the right degree of openness [29]. Besides the previously mentioned IoT stack [7], important work on business models also considered different revenue patterns in the area of iIoT [30]. Ehret and Wirtz identified a variety of potentials for IoT in the industrial application and concepts of iIoT business models [31]. Previous research has also discussed the appropriate organization structures and the required capabilities for non-standard partnerships and the make-or-buy decisions for iIoT platforms for manufacturing companies [32]. Some research also has identified iIoT related changes in business model elements [33]. Many research papers propose strategy frameworks, either for an integration in an existing IoT ecosystem [34], or for a classification of business models in IoT ecosystems including platforms [35]. Important work also explored of specific IoT platforms. Wortmann and Flüchter achieved a first classification of iIoT platforms [15]. Agarwal and Brem investigated the IT-enabled transformation of General Electric to an iIoT platform provider [36]. Sandberg et al have described the platform-based transformation of ABB [37]. Ardolino et al researched the capabilities for a successful service transformation in industrial companies [38].

Previous research on iIoT did not focus on the challenges of selecting the right platform from the perspective of a collaborative customer or a complementor. Accordingly, further research on concrete design criteria of iIoT platforms is required, addressing this challenge is required [15]. This paper fills this gap and extends the existing research in two directions. Firstly, the proposed taxonomy could extend the currently existing research on business model patterns for the growing area of iIoT platforms. Secondly, the proposed taxonomy provides a focused view on the machine tool industry in the iIoT and its characteristics of openness, which despite the increasing relevance of platforms, stays little investigated in the broad area of IoT.

3 Evaluating the business model criteria of iIoT platforms

3.1 Methodology

Qualitative research is suitable to analyze business decisions, which in our case was the decision for a certain iIoT platform. The database for this purpose contained primary data, which was obtained during qualitative interviews with practitioners. The interviews were conducted between March and August 2018 using a predefined interview guide and were thus semi-structured. The guide ensured the comparability, simultaneously offering enough freedom to create new specific or more in-depth questions, based on the answers. The interviews were compared and analyzed and the received information was recognized as single subjective dimensions of expert knowledge, which build a conceptualization and can be used for a theory generation [39]. As stated previously, data-driven maintenance was chosen as a platform-based service, to support the understanding of the interviewees, consequently defining the qualitative case study context. In the pre-selection process, suitable companies from the machine tool industry were identified based on publicly accessible company blogs, product presentations and press reports looking for digital services in the field of data-driven maintenance and related software-intensive services. The core target group consisted of mechanical engineering companies for various manufacturing processes in the metalworking, plastics processing and woodworking industries, as the initiators behind data-driven maintenance services. The interviewed representatives of the companies are specialized on processes such as milling, honing, turning, laser cutting and welding, injection molding wood construction joinery and others. An additional clustering of the identified companies includes machine makers, toolmakers, component makers and providers of automation solutions and software solutions for the automation or machine tools. Despite the heterogeneity of the processes and the companies, there are certain similarities between the studied companies. All these companies count as collaborative customers or complementors from the platform-provider perspective. At first, they all use iIoT platforms to build applications for software-intensive services as data-driven maintenance or similar. Consequently, the data-driven maintenance efforts of the studied companies and the applications built by them increase the overall value of the used iIoT platform and has impact on the iIoT ecosystem. The data collection process included interviews with machine tool companies (n=8), component suppliers including toolmakers, end effector manufacturers and automation solution providers (n=6), as well as manufacturing-related system integrators and consulting companies (n=3). The overall sample size consists of 17 interviews. After the evaluation of the 17th interview, the study has reached a theoretical saturation due to repetitive statements of the interviewees. The interviews were conducted with representatives working in the area or leading the digital service projects for their company’s products. The second requirement towards the representatives was to have at least 5 years of experience in their industry and in the digitization to ensure the qualification of the interviewees. The potential representatives were screened towards these two requirements, in order to count as experts on specific issues from the researcher's perspective [40]. The following table depicts the full list of interviewed experts during the data collection process of the study:

Table 1. Information on interviewed experts and their companies

ID	Position of the interviewee	Rounded no. of employees	Company profile
1	Head of Product & Services	50	Consulting and system integration
2	Product manager After Sales	350	Machine tools
3	Head of Industry 4.0 Campaign	7000	Components supplier
4	Head of Digitization	2000	Machine tools
5	Business Developer	800	Components supplier
6	Head of Maintenance	1300	Special machine tools
7	Managing Partner	10	Consulting
8	Corporate Innovation Management	900	Components supplier
9	Head of Technical Sales – E-conception	250	Machine tools
10	Technology manager Industry 4.0	2150	Machine tools
11	Head of industrial Data Services	500	Machine tools
12	Head of Development and Standardization Control	200	Components supplier
13	Head of Product Management	150	Machine tools
14	Head of Product Management	220	Components supplier
15	Lead Architect Industry 4.0	14000	Components supplier
16	Head of Product Management	70	System integration
17	Product manager Technical Support	11500	Machine tools

Predefined questions of the interview guide focused on the following topics:

- Which challenges of current importance do you experience during the implementation of data-driven maintenance?
- To what extent do you collaborate with partners during the implementation of data-driven maintenance?
- Which role do iIoT platforms take for data-driven maintenance?

The received information contained the project experience of the machine tool industry on iIoT platforms, including the challenges, the potentials and the value of the platform usage for data-driven maintenance and similar services. Hence, the data contains empirical evidence from companies about particular decisions on data-driven maintenance and iIoT platform selection and implementation, thus underlining interpretive research [41]. The analysis process of the recorded data included the transcription and coding processes of the interview recordings. During the coding process the answers were labeled, based on the interpretive identification of themes. The extraction of results underlies inductive reasoning [41], as the criteria and the characteristics are built from individual statements of the interviewed experts.

3.2 Building the taxonomy for iIoT platforms

The comparative analysis of coded transcripts returned five business model criteria for iIoT platforms. Each criterion can be aligned with the business model dimensions "How?" and "Value?" defined by Gassmann [21]. The first criterion provides a more detailed classification of platform openness and complies with the "How?" dimension. The taxonomy classifies this criterion in three additional characteristics:

- **Hardware integration openness:** While every iIoT platform mentioned by the interviewees was advertised as open, the least open iIoT platforms did not allow third-party application development at all. This means the business model of the iIoT platform provider also included the development of platform-based software. Openness on the other hand affects only the hardware integration. That means there are no strict exclusions of certain machine tools or electrical control components for process automation. Lastly, with this degree of openness the ecosystem can arise over the hardware components, as the platform provider develops the software-intensive services. The iIoT platform tapio, used in the wood working industry, currently shares this characteristic.
- **Project-related software integration openness:** This degree of openness allows external third-party development. The iIoT platform providers make the necessary resources for software development either available for a machine tool company (for its own IT department) or for an external system integrator on a project basis, a machine tool company can contract. The main distinctive feature of this certain degree of openness is that specific platform-based applications are developed in projects, without the orchestration of the integration or the distribution processes of the application through an app store by the platform provider. This degree of openness shares similar aspects as the supply-chain platform classification, shaped by Gawer [9]. However, the interviewed practitioners, who used an iIoT platform with this degree of openness, did not see any necessity in a further standardization in terms of an app store, due to the high specificity of their software-intensive services. Extending the hardware ecosystem, the software developing complementors can for instance be system integrators, either close to the machine tool company or to the platform provider company [28]. General Electric for instance shares this degree of openness for its platform Predix, maintaining a software ecosystem with

complementors for software development and integration [42], without the provision of an application store.

- **App store supported software integration openness**: This degree of openness means sharing of software development resources, consequently enabling external third-party development for a platform. The ecosystem evolves in terms of both hardware and software. Main distinctive features are the transparency of the service offerings and the standardization of applications driven by the app store. Though this degree of openness also requires checks and audits of complementors by the platform provider, the complementors can use the transparency of an app store for their advantage, for instance to screen it for missing software-intensive services. In addition, the machine tool companies can search for third-party partners for specific scenarios through the app store more precisely. That is why this degree of openness can be considered as the most open for a platform-based ecosystem. Siemens and SAP decided to share this degree of openness with their iIoT platforms Mindsphere and Leonardo, which are connected to enterprise application stores.

The next two identified characteristics concern the revenue stream of a platform provider and include the integration options and the revenue stream structure of the business model. As various iIoT platform providers also include the application development supplementary to the iIoT platform offering, they generate additional revenue streams, besides the infrastructure usage expenses. However, some platform providers offer free applications or pilot integration projects. The differences in the integration conditions belong in two dimensions of the Business Model Navigator: “How?” and “Value?”. The following list depicts the taxonomy:

- **Free integration:** In this context, it is important to understand the variety of strategies of provided iIoT platforms for the industrial application. There are some machine tool companies, which were able to introduce their own iIoT platforms and provide them within their industry. The interviewed representatives stated that the main goal of their company is to increase their end customer’s loyalty through additional value. The value is provided through iIoT platform-based applications for the machine tools, which are developed and integrated for free. The iIoT platform tapio for the wooden branch provides such integration conditions.
- **First integration free:** This integration allows the machine tool company to carry out a pilot use case without a financial risk. The first initial integration with a machine tool’s control unit and the development of an application are provided for free to lock-in the complementor on the iIoT platform and get additional revenue streams through the follow-up IoT projects. The Bosch IoT Cloud offers such an integration condition for the machine tool companies.
- **Paid integration:** This type of integration is different from the previous one, because the first application development is already a paid project. According to the interviewed representatives, Siemens offers this integration option for its iIoT platform Mindsphere.

Differing integration options also affect the revenue streams of an iIoT platform provider. The differing revenues belong in the “Value?” dimension of the Business Model Navigator. The taxonomy consists out of two characteristics, depicted below:

- **Indirect revenues:** The free integration generates additional indirect revenues in the business model of a platform provider through increased customer loyalty and access to customer’s specific problems in the production, consequently allow an improvement of the next generation of machine tools.
- **Direct revenues:** Integration conditions, which require direct payments for platform-based applications, whether from the beginning or from the second project on, generate direct revenue streams. Such a revenue structure differs significantly from the typical platform-based business models, which typically generate revenues through app store transactions or usage of infrastructure. These revenue streams differ from the typical platform-based business models for instance in the market for mobile OS.

The next two characteristics consider the differences in the service model architectures of the iIoT platforms. Although the iIoT platforms mostly seem as a PaaS model, an in-depth analysis reveals significant differences. Often, the cloud service model architecture of a focal iIoT platform is not evident from the perspective of a machine tool company. Nevertheless, this criterion plays an important role in the decision process for the right platform, as it has an impact on future partnerships of the machine tool company. Consequently, it affects different departments and lastly the whole platform-based iIoT ecosystem. The cloud hosting model complies with the “How?” dimension in the Business Model Navigator. The following list presents six most important out of eight characteristics of this criterion (see Fig. 1):

- **IaaS + PaaS:** This combination is mentioned separately due to its influence on the ecosystem growth. If the iIoT platform is bound to a predefined infrastructure provider, the machine tool company lacks the flexibility of provider change. Consequently, the vertical cooperation of the machine tool company with the infrastructure of choice and the ecosystem growth are restricted. If a machine tool company chooses for instance the Bluemix service by IBM it also uses IBM’s infrastructure.
- **PaaS + SaaS:** If the iIoT platform restricts third-party development and the platform provider is developing application in addition to the iIoT platform on its own, such a business model as a result restricts the horizontal cooperation of the machine tool company for instance with software development companies for future software-intensive services.
- **Partly IaaS + PaaS:** This type of cloud service model allows the machine tool company to choose, whether to buy the infrastructure additionally to the platform from the same provider or not. This optional offer extension could potentially restrict the selection of a third-party infrastructure partner and thus influencing the vertical ecosystem growth. Hewlett Packard Enterprise provides such a type of cloud service model.
- **PaaS + partly SaaS:** Some iIoT platforms as Mindsphere or Cumulocity allow third-party development. However, they also offer software development for their platforms by their own departments, competing with their business model in the

horizontal cooperation of a manufacturing company. That means applications could be developed by an external complementor or a platform provider. The platform provider could be more efficient in terms of adoption and integration of the application, while the complementor could have more knowledge about the specific process. Mindsphere app store represents this characteristic, as one can find there some basic applications developed by Siemens.

- **Partly IaaS + PaaS + partly SaaS:** This level of cloud services means that the iIoT platform can optionally be extended by the own infrastructure and application development, obtained from the iIoT platform provider. The machine tool company can decide about the restrictions, whether it chooses the full cloud computing stack from one source or not. SAP for instance shares this level of flexibility in the cloud service model for its iIoT platform Leonardo.
- **IaaS + PaaS + SaaS:** If the whole cloud computing stack is provided by one company, the iIoT platform business model restricts the horizontal and the vertical cooperation of a machine tool company. Bosch for instance offers the whole cloud computing stack, hosting its IoT Cloud on its own infrastructure and providing the implementation and the application development on their own.

Besides the cloud service model, the ability of iIoT platforms to be installed on-premise or support on-premise installations is also an important criteria for the machine tool companies. Connectivity and hosting possibilities were mentioned as an important criterion by many interviewed companies. This criterion is assigned to the "How?" dimension of the Business Model Navigator, divided as follow:

- **Cloud only:** This characteristic contains the iIoT platforms which are only hosted in the cloud. Additional connectivity modules could connect the iIoT platform with on-premise systems. However, the functionalities of the iIoT platform remain in the cloud. Most iIoT platforms typically provide this type of installation.
- **Hybrid installation:** This type of iIoT platforms allows an on-premise installation of modules and functionalities, if certain use case requires this. That means the iIoT platform is modularly divided between the cloud and the on-premise infrastructure. This type of installation is also commonly seen, as some functionalities or applications are installed in the edge and communication with the cloud, where historical data analysis is possible. Hybrid installations of iIoT platforms are commonly seen, if the platform provider offers additional hardware modules with certain pre-installed proprietary applications.
- **Possible on-premise installation:** This type of iIoT platforms allows to run the whole iIoT platform on-premise, if it meets the customer's requirements as an alternative to the cloud. This type of installation was a clear expressed requirement for some manufacturing companies and iIoT platforms such as edbic or Cumulocity allow this type of installation.

To sum up, the analysis of iIoT platform business models contains five criteria of iIoT platforms, extracted from single dimension statements of the interviewed experts. Each identified criterion is assigned to the dimensions "How?" or "Value?" of the Business Model Navigator [21]. The dimension "What?" represents the value proposition, which is regardless of the identified characteristics does not differ and has the goal to provide technologies for data-driven maintenance. This unifying dimension finding makes it possible to bridge different (economic and technical) criteria. Furthermore, each single characteristic of the taxonomy is assigned to at least one market-ready iIoT platform, additionally increasing the validity of the identified business model criteria, as it shows their occurrence on the market. The following structure of the identified criteria in a morphological box is the second artefact of this paper:

Dimensions	Business model taxonomy of industrial IoT platforms								
What?	Technologies for data-driven maintenance								
How?	Openness classification	Hardware integration openness – software development done only by the platform provider	Project-related software integration openness	App store supported software integration openness					
	Exemplary iIoT platform	tapio	Predix, Cumulocity	Mindsphere, Adamos					
How? Value?	Integration options	Free integration	First integration free	Paid integration					
	Exemplary iIoT platform	tapio	Bosch IoT Cloud	Leonardo, Azure					
Value?	Revenue streams	Indirect revenues	Direct revenues						
	Exemplary iIoT platform	Tapio	Mindsphere, Cumulocity						
How?	Cloud service model	IaaS	PaaS	IaaS + PaaS	PaaS + SaaS	Partly IaaS + PaaS	PaaS + partly SaaS	Partly IaaS + PaaS + partly SaaS	IaaS + PaaS + SaaS
	Exemplary iIoT platform	AWS	Predix, Thingworx	Bluemix, edbic	Tapio, FactoryTalk	HPE	Mindsphere, Cumulocity	Leonardo, Azure	Bosch IoT Cloud, Oracle IoT Cloud
How?	Connectivity and hosting	Cloud only	Hybrid installation	Possible on-premise installation					
	Exemplary iIoT platform	Azure, AWS	Leonardo, Bosch IoT Cloud, Mindsphere	Edbic, Cumulocity					

Fig. 1. Business model taxonomy of iIoT platforms

4 Conclusion

4.1 Findings and Limitations

This paper presents a taxonomy of iIoT platform criteria in the machine tool industry, based on dimensions of the St. Gallen Business Model Navigator and assigned to market-ready iIoT platform solutions. With the increasing relevance of the platform approach within the manufacturing industries, the identified criteria could help researchers and practitioners during further investigation of successful platform-based business models or suitable platform design in the iIoT. The demonstrated classification within the degree of openness could support the on-going benchmarking of the iIoT platforms, and by showing the differences supports the unanswered question about the right degree of openness and appropriate governance for manufacturing industries. Besides the classification of the openness degree, its interpretation by the potential complementors is even more important. The hardware integration openness may look as the least open alternative for iIoT platforms, but the interpretation by the complementors could be different. If for instance, the software integration openness provided by the app store is somehow restricted by the support of particular protocols or supports only platform-related proprietary standards and modules as certain programmable logic controller (PLC) systems are excluded on the hardware level, it may be the most closed alternative for a machine tool company at a second glance.

The morphological box forms a decision support tool for the important process of platform selection, which can be extended by additional platforms, not mentioned in the interviews. As the artefact contains economical and technical criteria, it could support heterogeneous stakeholders within a company, (for instance different departments), who are affected by the selection of a platform. In terms of the ISO 16355 the morphological box could assist the voice of the customer [43], providing a unifying artefact for affected stakeholders in different departments. Furthermore, manufacturing company at the early stage of entrance in the iIoT ecosystem could profit from the clearly assigned characteristics to market-ready platforms.

The morphological box features practical implications for platform providers to clarify their value propositions, because the criteria list represents the view of collaborative customers and complementors. In addition, the platform providers could use the morphological box to compare their iIoT platform against the competition and identify future niches for their branch of industry for the extension of their current offering.

Nevertheless, the results are limited, regarding the sample size of the qualitative interview study and the specific case study context (data-driven maintenance for machine tools) as the case study setting for IoT platforms. These limitations refer to the lack of generalizability of the findings. Furthermore, the interviews are subjects of subjective influence of the researcher and his understanding of the iIoT platform, thus forming the interview questions. As the conducted interviews were semi-structured, the follow-up questions, triggered by the answers of the practitioners could have led to an incomplete or wrong understanding of iIoT platforms – after all not every interviewed manufacturing company has already been using an iIoT platform for its software-intensive services. Some of the studied companies have developed their own software without using an iIoT platform and some companies have just managed to initiate pilot projects in the area of iIoT. Consequently, their knowledge on platforms could be limited, affecting the quality of the data sample.

4.2 Future research

The limitations of this paper require further research work on extension, generalization and evaluation of the taxonomy. The maturity of the platforms used in the cases has not been considered during this research, although the criteria evolvement during the platform lifecycle [44] could be a potential research area for a follow-up research. In addition, future research could also consider the sizes of the manufacturing companies as the collaborative customer and their impact on the selection of iIoT platforms and their criteria. A follow-up multiple case study analysis based on the taxonomy could also be a useful extension of the current result to check the completeness and the dependencies of the business mode criteria. During the given time, it was not possible to evaluate the taxonomy. Thus, future research should provide evaluation mechanisms, based on the performance of the utilized iIoT platforms for the manufacturing companies or on the impact of the identified criteria on the growth performance of the complete platform-based iIoT ecosystem.

Further research towards the customer's or complementor's interpretation of openness in the industrial application context is required. The interviews showed a varying and non-uniform understanding of platform openness from the practitioner's perspective. Moreover, the openness criteria of the taxonomy could also support a deeper research on optimal organizational capabilities of platforms in the field of iIoT and their interdependency with the identified criteria.

As the findings of this paper provide a conceptional base for a further research on iIoT platforms, a follow-up work should consider the platform governance. Especially an in-depth study of the currently used architecture and management of the application programming interfaces (API), software development kits (SDK) [45] and other boundary resources [46] in the field of iIoT platforms could make progress towards its impact on building an iIoT-platform based ecosystem.

Finally, the identified criteria could support the software-intensive business research on the development of new revenue streams for platform providers, beyond the ordinary pay-per-use models and traffic billing and have impact on the development of new business model patterns for iIoT.

References

1. Westkämper, E.: New paradigms of Advanced Manufacturing Engineering. In: Proceedings of the 17th International QFD Symposium, QFD Institute Deutschland e. V., Stuttgart (2011).
2. VDMA, McKinsey&Company: Zukunftsperspektive deutscher Maschinenbau, Study, McKinsey&Company (2014).
3. Zhong, R., Y., Xu, X., Klotz, E., Newman, S., T.: Intelligent Manufacturing in the Context of Industry 4.0: A Review. Engineering 3(5), 616-630 (2017).
4. Obermaier, R.: Industrie 4.0 als unternehmerische Gestaltungsaufgabe: Strategische und operative Handlungsfelder für Industriebetriebe. In: R. Obermaier (ed) Industrie 4.0 als unternehmeischer Gestaltungsaufgabe, pp. 3-34. Springer Fachmedien, Wiesbaden (2016).
5. Hoos, E., Gröger, C., Mitschang, B.: Mobile Apps in Engineering: A Process-Driven Analysis of Business Potentials and Technical Challenges. Procedia CIRP 33, 17-22 (2015).
6. Gröger, C., Silcher, S., Westkämper, E., Mitschang, B.: Leveraging Apps in Manufacturing. A Framework for App Technology in the Enterprise. Procedia CIRP 7, 665-669 (2013).
7. Porter, M., E., Heppelmann, J., E.: How Smart Connected Products are Transforming Competition. Harvard Business Review 92(11), pp. 64-8 (2014).
8. Gawer, A., Cusumano, M. A.: Platform leadership: How Intel, Microsoft, and Cisco drive industry innovation. Harvard Business School Press, Boston (2002).
9. Gawer, A.: Bridging differing perspectives on technological platforms: Toward an integrative framework. Research Policy 43, 1239-1249 (2014).
10. Turber, S., vom Brocke, J., Gassmann, O., Fleisch, E.: Designing Business Models in the Era of Internet of Things. Towards a Reference Framework. In: Proceedings of the International Conference on Design Science Research in Information Systems, pp. 17-31, Cham.

11. Werner, P. Petrik, D.: Design Implications for iIoT Platforms for the Machine Tools Industry. In: Proceedings of the 14. Wirtschaftsinformatik Conference (WI 2019). Forthcoming.
12. Hasselblatt, M.,Huikkola, T., Kohtamäki, M., Nickell, D.: Modeling manufacturer's capabilities for the Internet of Things. Journal of Business & Industrial Marketing (2018).
13. Krause, T., Strauß, O., Scheffler, G., Kett, H., Lehmann, K., Renner, T.: IT-Plattformen für das Internet der Dinge (IoT). Basis intelligenter Produkte und Services. Fraunhofer Verlag, Stuttgart (2017).
14. Herzwurm, G.: Position Statement. In: Brinkkemper, S., Abrahamsson, P., Maedche, A. and Bosch, J. (eds) Software Business, Platforms and Ecosystems: Fundamentals of Software Production Research. Report from Dagstuhl Seminar 18182, pp. 15. (2018).
15. Wortmann, F., Flüchter, K.: Internet of things. Business & Information Systems Engineering 57(3), 221-224 (2015).
16. Menon, K., Kärkkäinen H., Wuest, T., Gupta, J., P.: Industrial internet platforms: A conceptual evaluation from a product lifecycle management perspective. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 1-12 (2018).
17. Mechanical Engineering Industry Association Homepage, https://sud.vdma.org/viewer/-/v2article/render/26866475, last accessed 2018/10/03.
18. Kude, T.: Position Statement. In: Brinkkemper, S., Abrahamsson, P., Maedche, A. and Bosch, J. (eds) Software Business, Platforms and Ecosystems: Fundamentals of Software Production Research. Report from Dagstuhl Seminar 18182, pp. 12. (2018).
19. Peters, C., Blohm, I., Leimeister, J. M.: Anatomy of Successful Business Models for Complex Services: Insights from the Telemedicine Field. Journal of Management Information Systems 32(3), 75-104 (2015).
20. Täuscher, K., Laudien, S., M.: Understanding platform business models: A mixed methods study of marketplaces. European Management Journal 36(3), 319-239 (2018).
21. Gassmann, O., Frankenberger, K., Csik M.: The business Model Navigator. 55 Models That Will Revolutionise Your Business. Pearson, Harlow (2014).
22. Hartmann, M., Halecker, B.: Management of Innovation in the Industrial Internet of Things. In: Proceedings of the International Society for Professional Innovation Management (ISPIM), pp. 1-17. Budapest (2015).
23. BITKOM Homepage: Industrie 4.0 – Volkswirtschaftliches Potenzial für Deutschland. https://www.bitkom.org/noindex/Publikationen/2014/Studien/Studie-Industrie-4-0-Volkswirtschaftliches-Potenzial-fuer-Deutschland/Studie-Industrie-40.pdf, las accessed 2018/10/06.
24. Lee, E., A.: Cyber physical Systems: Design challenges. In: 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing, pp. 363-369. IEEE Computer Society, Washington C (2008).
25. Manyika, J., Chui, M., Bisson, P., Woetzel, J., Dobbs, R., Bughin, J., Aharon, D.: The Internet of Things: Mapping the value beyond the hype. McKinsey Global Institute (2015).
26. Lenz, J., Wuest, T., Westkämper, E.: Holistic Approach to Machine Tool Data Analytics. Journal of Manufacturing Systems 48(C), 180-191 (2018).
27. Engels, G., Plass, C., Rammig, F.-J.: IT-Plattformen für die Smart Service Welt. Verständnis und Handlungsfelder. acatech, München (2017).
28. Petrik, D., Herzwurm, G.: Stakeholderanalyse in plattformbasierten Ökosystemen für industrielle IoT-Plattformen. In: Proceedings of the Workshops of the German Software Engineering Conference 2019 (SE 2019). Forthcoming.
29. Mikusz, M., Csiszar, A.: CPS Platform Approach to Industrial Robots: State of the Practice, Potentials, Future Research Directions. In: PACIS 2015 Proceedings, pp. 176 (2015).

30. Weinberger, M., Bilgeri, D., Fleisch, E.: IoT business models in an industrial context. at-Automatisierungstechnik 64(9), 699-706 (2016).
31. Ehret, M., Wirtz, J.: Unlocking value from machines: business models and the industrial internet of things. Journal of Marketing Management 33, 111-130 (2017).
32. Bilgeri, D., Wortmann, F., Fleisch, E.: How Digital Transformation Affects Large Manufacturing Companies' Organization. In: Proceedings of the 38th International Conference on Information Systems, pp. 1-9. South Korea (2017).
33. Kiel, D., Arnold, C., Voigt, K.-I.: The influence of the Industrial Internet of Things on business models of established manufacturing companies – A business level perspective. Technovation 68, 4-19 (2017).
34. Smedlund, A., Ikävalko, H., Turkama, P.: Firm Strategies in Open Internet of Things Business Ecosystems: Framework and Case Study. In: Proceedings of the 51st Hawaii International Conference on System Sciences (HICSS), pp. 1591-1600 (2018).
35. Leminen, S., Rajahonka, M., Westerlund, M., Wendelin, R.: The future of the Internet of Things: toward heterarchical ecosystems and service business models. Journal of Business & Industrial Marketing 33(6), 749-767 (2018).
36. Agarwal, N., Brem, A.: Strategic business transformation through technology convergence: implications from General Electric's industrial internet initiative. International Journal of Technology Management 67, 196-214 (2015).
37. Sandberg, J., Holmström, J., Lyytinen, K.: Digital Transformation of ABB Through Platforms: The Emergence of Hybrid Architecture in Process Automation. In: Urbach, N., Röglinger, M (eds.) Digitalization Cases, pp. 273-291. Springer Nature, Cham (2019).
38. Ardolino, M., Rapaccini, M., Saccani, N., Gaiardelli, P., respi, G., Ruggeri, C.: The role of digital technologies for the service transformation of industrial companies. International Journal of Production Research, 2116-2132 (2017).
39. Bogner, A., Littig, B., Menz, W.: Interviewing Experts. Palgrave Macmillan, Basingstoke (2009).
40. Mieg, H., A., Näf, M.: Experteninterviews (2. ed.). Institut für Mensch-Umwelt-Systeme (HES), Zürich (2005).
41. Myers, M., D.: Qualitative Research in Business & Management (2. ed.). Sage, London (2013).
42. General Electric Homepage, https://www.ge.com/digital/partners, last accessed 2018/10/04.
43. ISO 16355-1: Application of statistical and related methods to new technology and product development process – Part 1: General Principles and Perspectives of Quality Function Deployment (QFD). ISO 16355-1:2015, Genf (2015).
44. Valenca, G., Alves, C., Jansen, S.: Strategies for managing power relationships in software ecosystems. The Journal of Systems and Software 144, 478-500 (2018).
45. Basole, R., C.: Position Statement. In: Brinkkemper, S., Abrahamsson, P., Maedche, A. and Bosch, J. (eds) Software Business, Platforms and Ecosystems: Fundamentals of Software Production Research. Report from Dagstuhl Seminar 18182, pp. 15. (2018).
46. Dal Bianco, V., Myllärniemi, V., Komssi, M., Raatikainen, M.: The Role of Platform Boundary Resources in Software Ecosystems: A Case Study. In: Proceedings of the IEEE/IFIP Conference on Software Architecture (WICSA), pp. 11-20. IEEE, Sydney (2014).

The Role of Prosumers in the Evolution of a Software Ecosystem

Case Steam

Tapani N. Joelsson[1], Sami Hyrynsalmi[2], and Sabine Molenaar[3]

[1] University of Turku, Department of Future Technologies, Turku, Finland
taneli@utu.fi
[2] Tampere University of Technology, Pervasive Computing, Pori, Finland
sami.hyrynsalmi@tut.fi
[3] Utrecht University, Department of Information and Computing Sciences, Utrecht, The Netherlands
s.molenaar@uu.nl

Abstract. This paper examines two questions: what is the role of active users or *prosumers*—i.e. users who consume as well as produce—in the transformation of a software ecosystem during its lifespan, and how does a digital marketplace transform into an ecosystem. This approach departs from the extant literature where consumers of an ecosystem are often treated only as passive participants. In this study, the role of prosumers is studied by portraying the transformation of *Steam*, by Valve Corporation, and discussed how it fits in the current field of software ecosystem research as well as what has been the impact of prosumers in its transformation process. The results from this case highlight the importance of users' active role in the evolution. Besides the discussion on the status of prosumers and the ecosystem evolution, the inner structure of this ecosystem is highlighted in the findings.

Keywords: Software ecosystem, Steam, prosumers, ecosystem evolution, ecosystem of sub-ecosystems, marketplace-driven ecosystem

1 Introduction

As the recent literature has often emphasised, different kinds of ecosystems have become the most discussed conceptualisation for understanding and explaining how the modern networked business world works. In the software industry, the 'software ecosystem' concepts and their descendants have become a liveable research field. Software ecosystem research seems to diverge into a few main groups [13, 40]; the absence of stability seems to be a common character for all kinds of ecosystems.

As pointed out already in the seminal paper by Moore [34], ecosystems—both natural and artificial—are not stable and they evolve through distinct phases over time. Therefore, it is not a surprise that ecosystem evolution has become a growing theme in the research of software ecosystems of all kinds. Yet, the number of studies assessing evolution of ecosystems remains low.

The previous work has addressed, e.g., evolution of open-source ecosystems [33, 37, 56], co-evolution of competing ecosystems [45, 58], as well as the transformation of a software product line into an ecosystem [2, 10]. To the authors' best knowledge, only Hanssen [10] has addressed how a system is transformed into an ecosystem with an empirical case.

Thus, this study answers the call of more empirical studies on this area and focuses, firstly, on the following research question:

RQ1 How does a digital marketplace transform into a software ecosystem?

In addition, the role of consumers in an ecosystem is often neglected. While their presence is acknowledged [12, 27, 29], they are often treated only as *'plankton'* [18] or as a source of consumer reviews and ratings [14]. Thus, there is a lack of studies on how the consumers are presented in these ecosystems. The driving research hypothesis for this study is that consumers are not passive entities in an ecosystem; in contrast, they might have a critical role in the evolution of an ecosystem.

Therefore, secondly, we address the following research question:

RQ2 What is the role of users in an evolution of a software ecosystem?

To answer the presented questions, we use a qualitative analysis of multivocal literature [c.f. 36] regarding *Steam* and its evolution. We document the birth and evolution of Steam, a digital distribution platform developed and operated by Valve Corporation (in the following 'Valve'). Valve was established in 1996 by Gabe Newell and Mike Harrington as a videogame enterprise. Since its initial inception Steam has evolved from an update tool for a few games to 'digital distribution platform' which at any given time has over 10 million concurrent users [49]. Valve does not reveal all the data, but figures used to estimate Steam's market share on the downloadable PC games market are usually around 50 to 70% [e.g. 9], market with an estimated value of around 27.1 billion dollars [1].

This study aims to answer to the request of more empirical analyses of different kinds of ecosystems and their evolution [e.g. 10, 17]. Based on the literature analysis of the software ecosystem literature by Manikas [27], Steam has until now remained a largely unstudied and unexplored software ecosystem. Furthermore, while marketplace-centred ecosystems, such as Google Play and Apple's AppStore for iOS devices, and their evolution have been researched previously, there is a lack of diversity in the research of this kind of ecosystems.

We attempt to show how it was not a single decision to become an ecosystem, but a series of conscious decisions, changes and afterthoughts in Steam and other factors around it that led to its current status as a definitive marketplace-driven ecosystem for PC gaming. As an example, along the way we will see how Steam itself and its components, not once but in several occasions, started their lives as specific tools for some purpose but have been expanded to more ambitious purposes later in their lives. In addition, this shows the role of consumers—or to be more exact, prosumers—in this evolution. The results contribute to the literature of software ecosystem evolution and diversification of the research by presenting an empirical analysis of a certain ecosystem and by bringing to light a previously underrepresented actor of the ecosystem.

The remaining of this paper is structured as follows. Section 2 discusses shortly on related work and the motivation behind this study. Section 3 presents details on the research approach used in this study and the rich description of the case subject is given in Section 4. It is followed by a discussion in Section 5. Finally, Section 6 concludes the study with some proposals for the directions of future work.

2 Background and Motivation

As the recent literature surveys have shown [e.g 27, 40], software ecosystems have become an active research topic in the computing discipline with hundreds of studies. While there are a dozen definitions of what constitutes an ecosystem [29], in this study we follow one of the most used and a classic definition by Jansen et al. [20]. According to this definition [20]:

> *"A software ecosystem is is a set of actors functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. These relationships are frequently underpinned by a common technological platform or market and operate through the exchange of information, resources and artifacts."*

As observed by several authors independently [19, 29], three repeating themes appear in most software ecosystem definitions—as well as in the aforementioned one. These are *i)* actors, *ii)* cooperation or business ecosystem, and *iii)* software. In our research subject, Steam, these requirements are fulfilled: *i)* Steam involves various actors, *ii)* there are relationships between the actors through the marketplace, and *iii)* Steam, naturally, involves shared software platforms and tools.

As pointed out by Manikas [27], the research field of software ecosystems is maturing and the research interests are diversifying. A research stream has arose to understand how software ecosystem emerge, evolve and transform. For example, Hanssen [10] presented a longitudinal case study on how a closed organisation with product line evolves towards an emerging ecosystem. However, this study concerns only a closed organisation.

Plakidas et al. [37] addressed the evolution of the R ecosystem and Teixeira & Hyrynsalmi [45] studied how several competitive ecosystems co-evolve. Yet, these studies are restricted to the evolution of already existing ecosystems and they do not address how an entity transforms into an emerging ecosystem.

Digital games are software products, and are therefore an important part of the software industry. The main difference between these from our perspective is the end use, general software is intended to be a tool or service used to fulfil a specific task. Contrary to this games are sold as entertainment, a voluntary action done for pleasure. [specifics at 22] Traditionally, distribution of games has been using the creator-publisher model, especially when games are distributed in physical format. Nowadays, with the advent of digital distribution, a growing number of games are published through digital stores independently by the creators, bypassing the publishers.

Steam was one of the first digital storefronts as it was announced in 2002 and published as public beta on January 2003. In example Apple's App Store and Android Market (now Google Play) were both opened during 2008 and Apple's iTunes added store-features on April 2003 [12]. Other stores specialised in games are for example GOG.com by CD Projekt[4] opened during 2008 and Origin by Electronic Arts[5] since 2011.

After this development story, we highlight the role of previously largely ignored ecosystem actors, consumers and especially the prosumers. In this paper, we follow a definition used previously on games and gamer related research by [57]. In this definition

> *"gamers who produce fan art, mods, or game-related materials to further contribute to the development of specific game titles can be thought of as "prosumers"."*

In this study gamers are the consumer base of Steam, and they have the possibility to have an active and influential role as prosumers. Their actions do have repercussions that shape or even revert the decisions made by the other actors in the network and the owner of the whole ecosystem.

3 Method

As noted, research papers related to Steam and its history turned out to be surprisingly few in numbers. In practice, this means that we found one Master's thesis by Shen [41] about the development of Steam, and an article by [21] with a short chapter about the history of Steam. Because of this we widened the net to include publications and interviews from gaming industry related news sources and other sources where these topics were discussed. Then again, as noted by [7] game industry related matters are in many cases reported in so-called "grey literature" instead of peer-reviewed academic publications.

During the data gathering period the non-academic sources were searched by using Google's search engine. Search strings were formed case by case. In the beginning they were broad and general (e.g. "valve + steam + history OR development") and refined for specific events or features (e.g. "valve + steam + workshop feature"). The academic references were sought by using search engines by ACM, IEEE and Google Scholar. As both, Valve and Steam are common words in the English language, all the searches used 'Valve' and 'Steam' together, but results still contained lots of papers from non-related fields. For this reason, the word 'game' was added to the base search string. Therefore, the basic searches were started with strings like "valve + steam + game + history" and "valve + steam + game + development". In both cases additional or clarifying sources were gathered by following citations and references.

The search criterion was to find sources where Steam's development was either the main topic or important milestones were reported and possibly clarified by people from Valve itself. Every possible branch of the story is not presented here, as our inclusion criterion dictated that we exclude material that has not attributed to the expanse of Steam. From this multivocal literature, we constructed the story presented shortly in Section 4.

[4] `https://www.cdprojekt.com/en/core-business/\#gog`
[5] `https://www.origin.com/fin/en-us/store/about`

4 Birth and Evolution of Steam

In the following, we will present the story of Steam so far through its five distinct phases. The first part presents Valve's development before Steam, the second tells Steams first-steps, the third describes Steam as a digital store and the fourth and fifth sections discuss Steam as an ecosystem. At the end of this section, a summary of the birth and evolution of Steam is presented in table 1.

4.1 Development of Valve before Steam (1996–2002)

The story of Steam starts with the foundation of its creator, Valve Corporation (originally Valve Software) in 1996 by Gabe Newell and Mike Harrington. In its early years, Valve concentrated on creating a multiplayer first-person shooter Half-Life, which they released in 1998. The distributor for this release was Sierra Entertainment and the core engine was licensed from id Software.

Importantly for their future, Half-Life included on its release also the level-design tool Worldcraft and software developer kit (SDK) for the players to create their own content and modifications. These tools led to popular modifications by fans, and in turn Valve hired them to turn the modifications to standalone games or purchased rights to work on them (e.g. Team Fortress Classic, 1999 and Counter-Strike, 2000).

During these early years updates for games were distributed as executable files that you could download from either the game creator itself or from a gaming related site that distributed them. This decentralised method of updating games led to situations where the player base was divided in groups that had different, incompatible versions of the game. This caused problems especially if an update was new or a game was updated several times during short time interval. Players with the wrong version of the game were not able to connect to the game servers, and this caused outcries and diminished the playing experience. [25]

To solve this problem Valve, whose games were popular multiplayer games, envisioned a tool that could be used to distribute updates for their own games, and also included additional features important for their games like anti-piracy and anti-cheat capabilities. During the process the digital store was added to the plan. They approached companies that had experience with creating similar kinds of network services (e.g. Microsoft, Yahoo! and RealNetworks), but they were turned down. Instead of abandoning the plan, they decided to create Steam by themselves. [25]

4.2 Steam as a Valve's Tool (2002–2004)

Steam was officially announced at the Game Developers Conference on March 22nd, 2002 by Gabe Newell[6]. In this initial announcement Steam was labelled as a broadband software delivery technology [54], and Valve's own titles were mentioned as the content, but that other service providers were already sought [44]. Beta testing for the new platform was conducted during early 2003[7] as a mandatory part of the beta program

[6] https://valvearchive.com/events/2002/GDC/
[7] http://counterstrike.wikia.com/wiki/Steam

for the then upcoming Counter-Strike 1.6. After this period Steam was released on September 12, 2003 [38].

An initial release on Steam was not mandatory for all games offered by Valve, but this changed in 2004 with the release of the much anticipated Half-life 2, sequel to their original hit [21]. During this launch Valve's infrastructure was unable to deal with the number of authentication requests and collapsed under the strain. As this authentication was required for all copies, even those bought on discs, gamers were not able to play the game even on a single player mode with the physical game discs on their computers' disc drive. [38] This rocky start of Steam and its performance problems did not give a good impression of itself to the gamers.

Within this period, all the games available on Steam were either created by Valve or Valve created them in cooperation with other studios. Additionally, during this period Valve also released their game engine Source for third-party developers. Valve's own games using this engine were popular and thus helped to spread Steam among the players, but Source was one of the first tools for game developers that also gained fame for Valve and their new service.

4.3 Steam as a Digital Game Store (2005–2009)

During this period Steam went through important milestones which would shape its future. First, in late 2005, other game developers started to sell and distribute their games through Steam. The very first non-Valve -related game on Steam was the Rag Doll Kung Fu by developer Mark Healey [11]. Other smaller teams and studios followed this and Steam started to gather momentum as the digital store for PC gamers.

The next major milestone for Steam's growth and emerging status came during 2007 when major developer-publisher studios like id Software and Eidos Interactive added their games to the catalogue [24]. Year 2007 also brought the "first-ever Steam storewide sale". This event was held between December 24th 2007 and January 1st 2008. [15, 46] Later on these sales would grow to become an anticipated event for the PC gaming crowd.

For third-party developers Valve launched the Steamworks on May 2008. Steamworks is a software development kit, a collection of tools and application programming interfaces, that allowed other developers to publish their own games in Steam without Valve being part of the integration phase. Before the Steamworks Valve had to be part of the process and act as a publisher for third-party games as they were the only one with access to the Steam's databases and other features [30].

Steamworks initial release eased the access to Steam and its features for third-party developers. Additionally Steamworks made it easier for them to implement digital rights management and crucial features for multiplayer games. This, in tandem with Steams rising in popularity in general, furhter increased other companies' interest in supporting the platform. Since its launch there have been several updates on Steamworks, for example during March of 2009 the support for downloadable content (DLC) and matchmaking were added to it. [47] Overall, during this period Steam solidified its status as the marketplace to be for the developers of PC games.

4.4 Steam as Software Ecosystem (2010–2014)

Steam had already been a community for gamers as the service connected them to each other and gave them a voice in the form of forums, chat features and by game reviewing. During this period gamers got more tools and opportunities from Valve to exercise their creativity and the voting power of their wallets.

Steam Workshops was added to Steam in October 2011[8]. With this addition users could create and share content they had created for games that could be expanded or modded by the users. In 2012 players got Steam Greenlight, a service with which they could decide by voting which games Valve would add to the Steam store. In the course of this year, non-gaming software was also added to steam, in a sense ending Steam's run as service only to the gamers. Family sharing features were added to enable content/game sharing among family members.

In March of 2013 the Steam Early Access[9] was launched. Under this service players could buy games that were still in various states of development and give feedback to the developers[10]. During 2014 the Steam Curators were introduced as part of the Discovery 1.0 update. The Discovery update's aim was to help buyers find games from Steam as the influx of games was making the process difficult by sheer volume. Curators are people or groups of people that make recommendations and reviews of games to other users of the Steam. [48, 50]

Signs of rising ambitions of Valve were shown as in 2012 they announced SteamOS, a Linux-based operating system and Steam Machines, console-type gaming devices for running it. The first prototypes of Valve's virtual reality headset were showcased during Steam Dev Days 2014. Steam Dev Days was supposed to be a yearly event, but so far it has been held only twice, during the years 2014 and 2016.

4.5 Steps beyond Software Ecosystem (2015–2018)

Up to this point Steam had been a platform for buying and playing games, a service that connected gamers to game developers and to each other. On the purely digital content delivery front, Steam expanded its offerings to include movies and television shows for streaming [e.g. 32]. Another feature catering for the gamers in Steam was the Steam Refund[11] service, which could be used to request refunds from purchases made through the Steam storefront.

But in the fourth quarter of 2015 the previously announced hardware projects started to materialise. SteamOS and Steam Machines, along with the Steam Controllers (gaming controllers) and Steam Link (digital media player for streaming Steam content to television sets) were released for consumers at this point. [51]

Valve's departure from digital content continued when they, in cooperation with HTC, developed the HTC Vive virtual reality headset [42]. For virtual reality Steam got a SteamVR extension and the Steamworks VR API was introduced. Valve also

[8] https://store.steampowered.com/news/16509/

[9] https://www.theverge.com/2013/3/20/4128644/steam-early-access-buy-and-play-games-still-in-development

[10] https://store.steampowered.com/earlyaccessfaq/

[11] https://store.steampowered.com/steam_refunds/

released the OpenVR software development kit to help VR content creators tackle the interoperability issues of various VR headsets. [55] During this period, one of the features added to Steam was the Steam Workshop which, Koch and Bierbamer [23] claim is Valve's *"attempt to create an ecosystem"*.

Table 1. Summary of the major milestones of and changes in Steam per phase.

Development of Valve before Steam (1996-2002)
Founding of Valve
Half-Life released
Decentralized method of updating games, resulting in divided player base
Plan for digital store
Steam as Valve's Tool (2002-2004)
Steam announced
Beta testing for the new platform
Steam released
Release of Half-Life 2
Performance issues lead to a rocky start
Release of Source game engine to third party developers
Steam as a Digital Store (2005-2009)
Other game developers start selling through Steam
Major developer-publisher studios add their games to Steam
"First-ever Steam storewide sale"
Launch of Steamworks, making Steam more accessible for third-party developers
Steamworks starts to support DLC
Steam as Software Ecosystem (2010-2014)
More tools and opportunities available for creativity and voting power
Steam Workshops was added for creation and sharing of content
Steam Greenlight introduced, a service for voting games into the store
Non-gaming software was added
Family sharing features were added
Steam Early Access was launched
Steam Curators were introduced
SteamOS was announced
Steps beyond Software Ecosystem (2015-2018)
Steam now includes movies and shows for streaming
Steam Refund service was added
Release of SteamOS for consumers
Release of Steam Machines for consumers
Release of Steam Controllers for consumers
Release of Steam Link for consumers
VR on Steam through the introduction of SteamVR and Steamworks VR API
Release of OpenVR SDK

5 Discussion

In the following, we will first discuss the key findings regarding the research questions of this study. It is followed by a discussion on the recent emergence of the 'ecosystem of ecosystems' and how these are manifested in Steam. The final subsections discuss the key observations and the limitations of this study.

5.1 RQ1: How does a digital marketplace transform into an ecosystem?

It almost seems that Steam's evolution has followed quite 'natural paths', emerged from the needs of the customers. That is, there was no visible plan to create Steam as an ecosystem from the fist day; however, through distinct phases it has evolved into its current shape. Steam started as an update tool, but as we have illustrated it has since grown far beyond its original scope. Many have already called it an ecosystem, like McElroy [31] even though he refers to a speech by Newell [35] where he is not using the term.

The same trend continues when Valve announced new additions to the service that they call the Ultimate Online Game Platform and the Ultimate Entertainment Platform [52]. Examples of this trend are the release of SteamOS, which by Dexter [5] was titled *"SteamOS Joins the Steam Ecosystem"* and in the case of HTC Vive, Gilbert [8] wrote how Valve is *"setting up an ecosystem with free tools for any company to use"*. However, occasionally, the term 'ecosystem' has been credited to Newell himself, such as by Statt [43] where it is stated that *"Newell has stressed that the point of the open-source philosophy behind Steam is not only to be as consumer- and community-friendly as possible, but also to build out the ecosystem as quickly and aggressively as possible."*

From a pure statistics viewpoint, as a distribution ecosystem, there has been aggressive growth. In the beginning of 2002 Steam started with one game from one company, at the end of 2017 it had 7,599 new released games and in the first six months of 2018 there are already over 4,600 new additions to the catalogue. This catalogue is serving a user/customer base of over 125 million users with a record of 18 million concurrent users.

Steam was born as an idea for updating software produced and sold by one company. Now we can look at it and see it as a multifaceted and multilayered ecosystem. On the one hand, Steam can be characterised as a monarchistic organisation where value is created by hybrid contributions distributed over a common platform [27]. On the other hand, a different kind of picture can be created by using vocabulary from [19]. Using their vocabulary, Steam is a privately owned software and service platform containing an extension market. From the accessibility viewpoint this market is either a screened market, but depending on role of participants it is either free or paid. Also, following Manikas et al. [28], we can see how Steam has started as a technological infrastructure which attracted other actors to join it. After this, it can be debated if it is a business-rooted or an actor-rooted ecosystem or hybrid of these two. Steam clearly has been an evolving and morphing system [33] during its existence, and seems to be continuing down this road.

Finally, it is worth to note that not all software ecosystems were just born as an ecosystem. While, for example, Google's Android operating system and its marketplaces were built as an ecosystem from the beginning, this study reported a different kind of story: the transformation of a software tool into an ecosystem. When compared to the transformation of a closed software product line company towards an open platform ecosystem company by Hanssen [10], similar distinct phases on the road to a software ecosystem can be identified. However, as Hanssen [10] focused on in-depth analysis of the company, his results also reflect the internal development of the case company. Whereas our focus is on Steam as an entity, thus our results reflect more on the development of the tool.

5.2 RQ2: The role of users

In the field of software ecosystems, consumers are excluded as they are the 'plankton' that keep the ecosystem alive as formulated by Jansen and Cusumano [18]. In other branches of ecosystem studies, consumers are noted [26], but their role and impact are not often at the centre of the studies.

In Steam, consumers are not voiceless or powerless. In Steam the plankton can become a 'modder' and provide extension(s) to a game using the Steam Workshop features. Also, they can participate in a game's success or failure by participating in their development while they are in the Early Access program. Other means of participation are the more traditional ways of writing reviews or becoming a curator who recommends games to others. So they are an active part of the ecosystem in several ways, not just by being the source for revenue to be shared by the business side of the ecosystem. Every consumer taking part in the Steam ecosystem has a voluntary chance to become a *prosumer* [39].

As mentioned, one way of showing their power is the usage of the reviewing feature [14]. This feature is intended for making (honest) reviews about games they are playing, so that other players can get guidance on whether they should buy some game or not. This is also how it is mostly used. However, gamers have also started to use it as a weapon, in the form of 'review-bombing'. In most cases this is a negative action where a large group of gamers rate a game negatively during a short period of time for some reason. For potential buyers this means that they will see that the game has lots of negative feedback and so the review-bombing makes the game less attractive for purchase. Valve has implemented fixes for this but the phenomenon has not been totally curbed[12].

The Early Access model introduced in 2013 has granted active consumers a way to participate in the game development process. This opportunity has since been seized by both the prosumers and the game development companies, as currently there are over 200 games offered for the player/prosumer communities through the Early Access service. Another new feature, the paid mods created by other gamers, that seems to be intended to embrace the prosumers received a different welcome. Instead of adoption, the community of gamers and modders rallied against paid content in the Workshop. For example on change.org, 132,458 voters petitioned for the removal of this feature, and in this case gamers won. Valve (and in this case Bethesda, another gaming company)

[12] E.g. `https://www.theverge.com/2017/9/20/16336290/valve-steam-review-bomb-charts-abuse-update`

jointly removed the feature.[13,14] These events are covered more in-depth in the article by Joseph [21].

As a final note, the role and involvement of consumers in software ecosystems should be studied more. Previously, consumers have been mainly seen adding value to the ecosystem by verbal or numerical reviews [e.g. 14] or as the aforementioned 'plankton' that keeps the ecosystem alive by providing the financial food to it. The case of Steam shows that the role of consumers—or prosumers—can be more important than previously presented in the software ecosystem literature. This might be due to the fact that games are entertainment software, which gamers use voluntarily to have fun. Bluntly this means that for the gamers it is not enough to just get the software(game) as it is in their role during working hours as workers using productivity software. If, as gamers, they are not getting value for their money and time, they will complain in reviews and on other forums, or use the Steam Refund service to get their money back instead of blowing through the game like they might be doing during working hours with productivity software. This also opens interesting research and development avenues for other kinds of ecosystems, especially in case of marketplace-driven ecosystems.

5.3 Ecosystem made of ecosystems

The digital distribution service role of Steam is the backbone and the root of the ecosystem created by Valve, with estimated profits in *"high number hundreds of million dollars"*[15]. This part of the ecosystem is partly for business-to-business and partly for business-to-consumers transactions. All earnings always involve Valve as everybody pays a fee to Valve for their sales inside the ecosystem. Revenue to the third party companies, and to the other content producers, comes from the sales to other companies or to the customers. There also exist sub-ecosystems inside the overall Steam ecosystem. These can be divided into several classes. There are the per game 'bubble-ecosystems' created around individual games where creative gamers have created new content or mods to some popular game using the available tools and Steam Workshop features. Then, some games have their own internal economies and virtual currencies, creating another layer of ecosystems. For example, Team Fortress 2 and Counter-Strike: Global Offensive have microtransactions for cosmetics items. In some of these games players can also trade items they have gathered, which has created in-game markets and in some cases third party market sites. There are also hardware-specific and bound ecosystems inside Steam, like Steam VR, which is bound to virtual reality hardware and APIs, thus being separated from the general PC entertainment software available in Steam.

Other bubbles are formed around the productivity software sold in and distributed through Steam. These could be seen as business to business ecosystems coexisting with gaming software, which are part of the business to consumers ecosystem.

---

[13] https://www.change.org/p/valve-remove-the-paid-content-of-the-steam-workshop

[14] http://steamcommunity.com/games/SteamWorkshop/announcements/detail/208632365253244218

[15] https://www.forbes.com/forbes/2011/0228/technology-gabe-newell-videogames-valve-online-mayhem.html

There is interaction between these separate layers and bubbles. The big question for Valve has been how to control these bubbles and the virtual economy in the overall Steam ecosystem. To tackle some of these issues, Valve hired economist Yanis Varoufakis as the economist-in-residence [53]. As a final note, the field of "ecosystem of ecosystem" is, to the authors' best knowledge, still mainly uncovered area in the software ecosystem research and future work is needed.

5.4 Key observations, limitations and future work

We recapitulate our key observations in the following:

1. Users have had a clear role in the evolution of the Steam ecosystem. This contradicts the extant literature, which often understates the role of consumers in the ecosystems. While Steam as a 'video game ecosystem' might differ with its key characteristics from other kinds of ecosystems [c.f. 13], this, nevertheless, hints that consumers as active participants – i.e., prosumers – of an ecosystem should be addressed more.
2. The evolution story of Steam has followed quite 'natural paths', which enriches our understanding of the transformation and birth of software ecosystems. While the previous studies have reported the transformation starting from technical changes [10, 37, 45] as well as from the customers' requirements [10], no major external changes were seen driving the transformation. Yet, more qualitative studies are needed to understand the internal rationalities behind the change.
3. Finally, this study also notes the emergence of 'ecosystem bubbles' inside the Steam ecosystem. While there is a growing interest towards ecosystem of ecosystems [16], these still remain an underresearched area. However, future work is needed to better understand the dynamics of these kinds of multilayered ecosystems.

As with all studies, there are certain limitations restricting the validity of this study and generalisation of the results. First and foremost, we are looking at Steam and its history as outsiders, relying on clues and scraps of information coming from third-party sources. As so, it might be that we are missing some parts of the story and, thus, future work should verify these results as well as add more details by interviewing the personnel related to the development of Steam.

Secondly, generalisation of this study is remarkably limited to this kind of an entity. Steam is an interesting research subject due to its popularity, being the first and the largest gaming ecosystem; however, it is hard to generalise from such a case.

In addition to the already proposed issues for further studies, Steam also offers an interesting study subject to understand the value creation and caption in an ecosystem. Thus, by studying Steam, a more comprehensive picture of ecosystem value creation mechanisms could be created. Furthermore, Steam and its rival gaming ecosystems could also serve as case study subjects for further studies aiming to understand the competition between ecosystems. For example, Valve's games currently cannot be found from, e.g. GOG.com by CD Projekt, while some of CD Projekt's games can be found on Steam.

Also, in general, marketplace-driven ecosystems offer an interesting cases for the study of how the power of different actors are manifested on them. For example, in the case of Steam it is privately owned and solely under control of the mother company Valve and its decisions, but so far they seem to have been distributing their power to other actors, e.g. the power to accept new products into the ecosystem has been moved from Valve to consumers and then from this player curated model to simpler paid-entry.

Another future direction would be to observe and study how ecosystems respond to an emerging competition. In case of Steam, there are, for example, two emerging competitors in the form of Discord and Tencent. Discord is a popular communication app with 130 million users [4] and Tencent is well-known Chinese gaming company with a reported total revenue of $22 billion and owner of WeGame, a digital video game store and social platform [3]. Both of these examples are currently reported in the industry media as potential rivals for Steam's current dominance in PC gaming as they are pivoting their operations [4, 6].

6 Conclusion

In this work, we documented the birth and evolution of a gaming software ecosystem: Steam by Valve Corporation. Steam started its life as a digital tool to distribute digital updates for games. Currently Steam does that and has also crossed from totally digital phenomena to the physical side with related hardware products. Steam can be easily described as a software ecosystem and this study discussed the distinct evolution phases of it and especially how the end-users have their own active and power-wielding position inside it. The study contributes to the field of software ecosystem research by responding to the call of more work on different kinds of software ecosystems. Furthermore, the case of Steam emphasises the need to discuss the role of consumers in the software ecosystems more, as well as to study the emergence of 'ecosystem of ecosystems' in large ecosystems.

References

[1] Batchelor, J.: Gamesindustry.biz presents... the year in numbers 2017 (2017), available at: https://www.gamesindustry.biz/articles/2017-12-20-gamesindustry-biz-presents-the-year-in-numbers-2017

[2] Bosch, J.: Maturity and evolution in software product lines: Approaches, artefacts and organization. In: Chastek, G.J. (ed.) Proceedings of the Second International Conference on Software Product Lines (SPLC 2). Lecture Notes in Computer Science, vol. 2379, pp. 257–271. Springer Berlin Heidelberg, London, UK (2002). https://doi.org/10.1007/3-540-45652-X_16

[3] Chalk, A.: Chinese gaming giant tencent is rebranding its digital store as 'wegame' (2017), available at https://www.pcgamer.com/chinese-gaming-giant-tencent-is-rebranding-its-digital-store-as-wegame/

[4] Chapple, C.: Could discord dethrone steam as the go-to pc games marketplace? (2018), available at https://www.pcgamesinsider.biz/interviews-and-opinion/67361/could-discord-dethrone-steam-as-the-go-to-pc-games-marketplace/

[5] Dexter: Steamos joins the steam ecosystem (2013), available at: http://geekbeat.tv/steamos-joins-the-steam-ecosystem/

[6] Donnelly, J.: Tencent plans to take its steam-like wegame store global (2018), available at https://www.pcgamer.com/tencent-plans-to-take-its-steam-like-wegame-store-global/

[7] Engström, H., M.B.B.P., Toftedahl, M.: Game development from a software and creative product perspective: A quantitative literature review approach (2018). https://doi.org/10.1016/j.entcom.2018.02.008

[8] Gilbert, B.: Valve's push into VR will span many headsets from many companies (2015), available at: https://www.engadget.com/2015/03/04/valve-steam-vr-initiative/

[9] Grubb, J.: Valve won't manually curate steam because it dominates pc gaming (2017), available at: https://venturebeat.com/2017/02/13/valve-wont-manually-curate-steam-because-it-dominates-pc-gaming/

[10] Hanssen, G.K.: A longitudinal case study of an emerging software ecosystem: Implications for practice and theory. Journal of Systems and Software **85**(7), 1455–1466 (July 2012). https://doi.org/10.1016/j.jss.2011.04.020

[11] Healey, M.: Rag doll kung fu at steam store (2018), available at: http://store.steampowered.com/app/1002/Rag_Doll_Kung_Fu/

[12] Hyrynsalmi, S.: Letters from the War of Ecosystems — An Analysis of Independent Software Vendors in Mobile Application Marketplaces. Doctoral dissertation, University of Turku, Turku, Finland (December 2014). https://doi.org/10.13140/2.1.4076.4484, http://urn.fi/URN:ISBN:978-952-12-3144-5, TUCS Dissertations No 188

[13] Hyrynsalmi, S., Ruohonen, J., Seppänen, M.: Healthy until otherwise proven: some proposals for renewing research of software ecosystem health. In: Proceedings of the first software health workshop. pp. 18–24. ACM (2018)

[14] Hyrynsalmi, S., Seppänen, M., Aarikka-Stenroos, L., Suominen, A., Järveläinen, J., Harkke, V.: Busting myths of electronic word of mouth: The relationship between customer ratings and the sales of mobile applications. Journal of Theoretical and Applied Electronic Commerce Research **10**(2), 1–18 (May 2015). https://doi.org/10.4067/S0718-18762015000200002

[15] Internet Archive: State of steam frontpage on dec 29th 2007 (2007), available at: https://web.archive.org/web/20071229021148/http://steampowered.com/v/index.php

[16] Jansen, S.: Introduction to the keynote on software ecosystem governance. In: Hyrynsalmi, S., Suominen, A., Jud, C., Bosch, J. (eds.) Proceedings of the 9th International Workshop on Software Ecosystems. CEUR Workshop Proceedings, vol. 2053, pp. 1–2. CEUR-WS, Aachen, Germany (Nov 2017)

[17] Jansen, S., Brinkkemper, S., Finkelstein, A.: Business network management as a survival strategy: A tale of two software ecosystems. In: Jansen, S., Brinkkemper, S., Finkelstein, A., Bosch, J. (eds.) Proceedings of the first International Workshop on Software Ecosystems. CEUR Workshop Proceedings, vol. 505, pp. 34–48. CEUR-WS (September 2009), http://ceur-ws.org/Vol-505/iwseco09-5JansenBrinkkemperFinkelstein.pdf

[18] Jansen, S., Cusumano, M.A.: Defining software ecosystems: A survey of software platforms and business network governance. In: Jansen, S., Bosch, J., Alves, C.F. (eds.) Proceedings of the Fourth International Workshop on Software Ecosystems. CEUR Workshop Proceedings, vol. 879, pp. 41–58. IWSECO, CEUR-WS, Cambridge, MA, USA (June 2012), http://ceur-ws.org/Vol-879/paper4.pdf

[19] Jansen, S., Cusumano, M.A.: Defining software ecosystems: a survey of software platforms and business network governance. In: Jansen, S., Brinkkemper, S., Cusumano, M.A. (eds.) Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry, chap. 1, pp. 13–28. Edward Elgar Publisher Inc., Northampton, MA, USA (2013). https://doi.org/10.4337/9781781955635.00008

[20] Jansen, S., Finkelstein, A., Brinkkemper, S.: A sense of community: A research agenda for software ecosystems. In: 31st International Conference on Software Engineering — Companion Volume, ICSE-Companion 2009. pp. 187–190. IEEE (May 2009). https://doi.org/10.1109/ICSE-COMPANION.2009.5070978

[21] Joseph, D.J.: The discourse of digital dispossession: Paid modifications and community crisis on steam. Games and Culture **0**(0), 1555412018756488 (0). https://doi.org/10.1177/1555412018756488, https://doi.org/10.1177/1555412018756488

[22] Kasurinen, J.: Games as software: Similarities and differences between the implementation projects. In: Proceedings of the 17th International Conference on Computer Systems and Technologies 2016. pp. 33–40. CompSysTech '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2983468.2983501, http://doi.acm.org/10.1145/2983468.2983501

[23] Koch, S., Bierbamer, M.: Opening your product: impact of user innovations and their distribution platform on video game success. Journal of Electronic Markets **26**, 357–368 (2016)

[24] Kuorikoski, J.: Finnish Video Games: A History and Catalog. McFarland (2015)

[25] Lee, J.: The last of the independents? (2008), gameindustry.biz Available at: http://www.gamesindustry.biz/articles/the-last-of-the-independents-

[26] Litovuo, L., Makkonen, H., Aarikka-Stenroos, L., Luhtala, L., Makinen, S.: Ecosystem approach on medical game development: The relevant actors, value propositions and innovation barriers. In: Proceedings of the 21st International Academic Mindtrek Conference. pp. 35–44. AcademicMindtrek '17, ACM, New York, NY, USA (2017). https://doi.org/10.1145/3131085.3131104, http://doi.acm.org/10.1145/3131085.3131104

[27] Manikas, K.: Revisiting software ecosystems research: A longitudinal literature study. Journal of Systems and Software **117**, 84–103 (July 2016). https://doi.org/10.1016/j.jss.2016.02.003

[28] Manikas, K., Hämäläinen, M., Tyräinen, P.: Designing, developing, and implementing software ecosystems: Towards a step-wise guide. In: Proceedings of the International Workshop on Software Ecosystems co-located with 10th International Conference on Information Systems (ICIS 2016). CEUR Workshop Proceedings, vol. 1808, pp. 70–79. CEUR-WS (2016)

[29] Manikas, K., Hansen, K.M.: Software ecosystems — A systematic literature review. Journal of Systems and Software **86**(5), 1294–1306 (May 2013). https://doi.org/10.1016/j.jss.2012.12.026

[30] Martin, J.: Valve releases Steamworks SDK (2008), available at: http://www.bit-tech.net/news/gaming/valve-releases-steamworks-sdk/1/

[31] McElroy, G.: Gabe newell on valve's past, gaming's present and steam's future (2013), available at: https://www.polygon.com/2013/2/1/3941274/gabe-newell-steam-box-talk-ut

[32] Meer, A.: Oh good, the leprechaun films are on steam (2016), available at: https://www.rockpapershotgun.com/2016/04/29/steam-movie-streaming/

[33] Monteith, J., McGregor, J., Ingram, J.: Hadoop and its evolving ecosystem. In: Proceedings of the 5th International Workshop on Software Ecosystems. CEUR Workshop Proceedings, vol. 987, pp. 57–68. CEUR-WS (2013)

[34] Moore, J.F.: Predators and prey: A new ecology of competition. Harvard Business Review **71**(3), 75–86 (May-June 1993)

[35] Newell, G.: Gabe newell: Reflections of a video game maker. youtube recording of talk at lyndon b. johnson school of public affairs (2013), available at: https://www.youtube.com/watch?v=t8QEOBgLBQU

[36] Ogawa, R., Malen, B.: Towards rigor in reviews of multivocal literatures: applying the exploratory case study method. Review of Educational Research **61**, 265–286 (1991)

[37] Plakidas, K., Stevanetic, S., Schall, D., Ionescu, T.B., Zdun, U.: How do software ecosystems evolve? a quantitative assessment of the r ecosystem. In: Proceedings of the 20th International Systems and Software Product Line Conference. pp. 89–98. SPLC '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2934466.2934488, http://doi.acm.org/10.1145/2934466.2934488

[38] Plunkett, L.: Steam is 10 today. remember when it sucked? (2013), available at https://kotaku.com/steam-is-10-today-remember-when-it-sucked-1297594444

[39] Ritzer, G., Jurgenson, N.: Production, consumption, prosumption: The nature of capitalism in the age of the digital 'prosumer'. Journal of consumer culture **10**, 13–36 (2010)

[40] Seppänen, M., Hyrynsalmi, S., Manikas, K., Suominen, A.: Yet another ecosystem literature review: 10 + 1 research communities. In: 2017 IEEE European Technology and Engineering Management Summit (E-TEMS). pp. 1–8. E-TEMS 2017, IEEE (Oct 2017). https://doi.org/10.1109/E-TEMS.2017.8244229

[41] Shen, D.Z.: The people's history of Steam. Master's thesis, Georgetown University (2015)

[42] Souppouris, A.: How htc and valve built the vive – a vr headset four years in the making (2016), available at: https://www.engadget.com/2016/03/18/htc-vive-an-oral-history/

[43] Statt, N.: Valve's steam machine lineup poses massive threat to gaming status quo (2014), available at: https://www.cnet.com/news/valves-steam-machine-lineup-poses-massive-threat-to-gaming-status-quo/

[44] Sulic, I.: Gdc 2002: Valve steams. 22.3.2002 (2002), available at: http://www.ign.com/articles/2002/03/22/gdc-2002-valve-steams
[45] Teixeira, J., Hyrynsalmi, S.: How do software ecosystems co-evolve? a view from openstack and beyond. In: Ojala, A., Holmström Olsson, H., Werder, K., Ojala, A., Holmström Olsson, H., Werder, K. (eds.) Software Business: 8th International Conference, ICSOB 2017, Essen, Germany, June 12-13, 2017, Proceedings. pp. 115–130. No. 304 in Lecture Notes in Business Information Processing, Springer International Publisher, Cham (2017). https://doi.org/10.1007/978-3-319-69191-6_8
[46] Truta, P.: 10-50% off for all steam games! (2007), available at: http://news.softpedia.com/news/10-50-Off-for-All-Steam-Games-74785.shtml
[47] Valve Corporation: Steamworks makes drm obsolete (2009), press release by Valve, available at: http://store.steampowered.com/news/2372/
[48] Valve Corporation: The discovery update (2014), available at: http://store.steampowered.com/about/newstore
[49] Valve Corporation: Steam & game stats (2018), available at: http://store.steampowered.com/stats/
[50] Valve Corporation: Steam curators – curated shopping with the voices you trust (2018), available at: http://store.steampowered.com/about/curators/
[51] Valve Corporation: Steam hardware (2018), available at: http://store.steampowered.com/hardware/
[52] Valve Corporation: Steam, the ultimate online game platform (2018), available at: http://store.steampowered.com/about/
[53] Varoufakis, Y.: It all began with a strange email (2012), available at: http://blogs.valvesoftware.com/economics/
[54] Walker, T.: Gdc 2002: Valve unveils steam (2002), available at: https://www.gamespot.com/articles/gdc-2002-valve-unveils-steam/1100-2857298/
[55] Wawro, A.: Valve launches new OpenVR SDK to expand SteamVR development (2015), available at: https://www.gamasutra.com/view/news/242401/Valve_launches_new_OpenVR_SDK_to_expand_SteamVR_development.php
[56] West, J., Wood, D.: Evolving an open ecosystem: The rise and fall of the Symbian platform. In: Adner, R., Oxley, J.E., Silverman, B.S. (eds.) Collaboration and Competition in Business Ecosystems, Advances in Strategic Management, vol. 30, pp. 27–67. Emerald Group Publishing Limited (2013). https://doi.org/10.1108/S0742-3322(2013)0000030005
[57] Wu, H.: Video game prosumers: Case study of a minecraft affinity space (2016). https://doi.org/doi:10.5406/visuartsrese.42.1.0022
[58] Yu, L., Ramaswamy, S., Bush, J.: Software evolvability: An ecosystem point of view. In: Third International IEEE Workshop on Software Evolvability. pp. 75–80. IEEE (2007). https://doi.org/10.1109/SE.2007.8

Entrepreneurial Oriented Discussions in Smart Cities: Perspectives Driven from Systematic Use of Social Network Services Data

Arash Hajikhani

Innovations, Economy, and Policy, VTT Technical Research Centre of Finland
arash.hajikhani@vtt.fi

Abstract. The concept of the "smart city" has become popular in scientific literature and international policies in the past two decades. Smart cities are known as a system of physical infrastructure, the ICT infrastructure and the social infrastructure exchanging information that flow between its many different subsystems. The "smart cities" concept has been introduced with various dimensions among those, the embedded ICT infrastructure in smart cities is playing a decisive role among the functions of the system. One of the important derivatives of ICT is the new communication mediums known as Social Network Services (SNSs) which is emerging and introducing additional functionalities to "smart cities". This paper seeks to advance the understanding of SNSs in smart cities for evaluating the effects on the innovation and entrepreneurial ecosystem. This agenda has been tackled by a rigorous methodological approach in order to capture and evaluate the presence of entrepreneurial oriented discussion in a popular SNSs medium (Twitter).

Keywords: Smart Cities, Social Network Services, Start-ups, Content Analysis.

1 Introduction

Population growth and the urbanization associated to that are recognized as the contemporary challenges that seeks novel, efficient, effective, and economic approaches to better governance. Challenges for developing the infrastructures and services needed to be addressed so to increase communities living standards. The emergence of the "smart city" concept can be considered as a response to such challenges ensuring that cities can develop economically, whilst protecting the environment and quality of life for citizens. Smart technologies is offering cities exciting possibilities for the provision of new services and integrated city infrastructures, as well as supporting innovation, digital entrepreneurship, and sustainable city development [10]. According to World Economic Forum [47], a growing number of cities around the world are implementing ambitious smart city programs and projects across a range of themes including governance, local economic development, citizen participation, urban living, the natural and built environment, and sustainable transport.

An in-depth analysis of the existing literature revealed that the smart city is a multi-faceted concept with many elements and dimensions. Descriptions of smart cities are now including qualities of people and communities as well as ICTs. The smart cities are known as a system of physical infrastructure, the ICT infrastructure, and the social infrastructure exchanging information that flows between its many different subsystems [2]. It might even be noticeable that major cities can serve as a good representation of a nation's economic success or failure. According to Beattie [4] that's because the tricky business of development and urbanization can play a big role in a country's economic prosperity. Entrepreneurship and innovation is the major concern for an economy consequently within the boundary of a city therefore, the competitiveness of a city today is determined by its innovativeness and economic strength [3]. While researchers have realized that smart cities are more entrepreneurial than others [28,34], an analysis of the detailed characteristics accounting for this higher entrepreneurial activity within smart cities has not been conducted.

One of the major resources connected to the success of smart cities is the societal capital or cultural capital within the city boundaries. The emphasis on the role of social capital in urban development is promoted in parallel to technical aspects of a city [25]. The importance of human and social capital has been recognized by smart city definitions from previous literature, and it has been seen as a fundamental aspect of any smart city [2,11,27,40]. Social capital has also been seen as an important dimension for facilitation of innovation and entrepreneurship in smart cities. Smart cities have the infrastructure to bridge and facilitate the connectivity of society for entrepreneurial activity. Despite the recognition of the importance of the human and social capital aspect in smart cities, the measurement and assessment of this aspect has remained a challenge. Performance measurement studies on smart cities dimensions, especially on social and human capital, are subject to being outcome indicators that, by their nature, involve medium- to long-term observation and detection times [30]. The results of this issue are the lack of insight coming from society and incapability to absorb the information coming from society.

In this research, the attempt is to study the smart city social and human capital performance measurement concerning innovation and entrepreneurship oriented activity. Due to ICT advancements, smart cities have the infrastructure to bridge and facilitate the connectivity of society. Within the broad spectrum of ICT application, the emerging presence of the mass media communications such as Social Network Services (SNSs) and social media has not been taken into account for studying innovation and entrepreneurship ecosystem in smart cities. Publicly available data sources such as Twitter have facilitated massive data collection which can leverage the research at intersection of social sciences, data sciences, and indicator design, thus informing the research community of major opinions and topics of interest among the general population [45,48] that cannot otherwise be collected through traditional means of research (e.g., surveys, interviews, focus groups) [17]. On the other hand, citizens are empowered to use technology oriented common platform to communicate among themselves, which resulted in inclusive use of social network services among citizens. Yet despite this interest, there seems to be very limited understanding of what the "social networking services" or "social media" exactly represent and do to societies. In our presented case, we saw social media discussion as a curtail pillar in regulating entrepreneurial oriented discussions in smart cities. Therefore, this paper explores the social network services role in smart cities from the innovation and entrepreneurial ecosystem vantage point. We aim to address the following research questions:

- How can smart cities leverage the presence of SNSs for entrepreneurial oriented activities in innovation ecosystem?
- Utilize social network services data to identify the presence of impactful entrepreneurial discussion (a methodological approach).

This agenda has been tackled by a rigorous methodological approach in order to capture and evaluate the presence of entrepreneurial oriented discussion in a popular SNSs outlet (Twitter). A thorough process of detecting and capturing relevant tweets was performed to evaluate the usage of SNSs in promoting innovation and entrepreneurial oriented discussions. Based on the recognized Smart City Index, London city has been selected to utilize the methods for capturing social capital on innovation and entrepreneurial activity.

2 What are smart cities?

Cities are considered as key role players in social and economic aspects in global perspectives, and therefore in order to understand the importance of cities as future key elements, the definitions of "smart cities" will be explored in this section. United Nations Population Fund indicates that in the year 2008 about 3.3 billion people, which is more than 50 percent of global population, lived in urban areas. This estimation is expected to increase to 70 percent by 2050 according to a United Nations report [44]. The urbanization figure in Europe is currently 75 percent of the population and the number is expected to reach 80 percent by 2020 [44].

The advantage point of smart cities as a structure to enable the pre mentioned movements has been seen on the opportunity for information exchange that flows between its many different subsystems [20]. A comprehensive definition of smart cities by Nijkamp and Kourtit [33] "Smart cities are the result of knowledge-intensive and creative strategies aiming at enhancing the socio-economic, ecological, logistic and competitive performance of cities. Such smart cities are based on a promising mix of human capital (e.g. skilled labor force), infrastructural capital (e.g. high-tech communication facilities), social capital (e.g. intense and open network linkages), and entrepreneurial capital (e.g. creative and risk-taking business activities)". Hence, a recent classification by Neirotti et al. [32], define two major domains for the smart city concept with regard to the exploitation of tangible and intangible urban assets: (1) hard domain, which concerns energy, lighting, environment, transportation, buildings, and health care and safety issues and (2) soft domain, which addresses education, society, government, and economy. Shapiro [36] and Holland [24] argue over soft domain aspect of smart cities such as human capital rather that hard domain aspects like ICT; as the driver of smart city creation. According to Caragliu et al. [11] a city is smart "when investments in human and social capital and traditional (transport) and modern (ICT) communication infrastructure fuel sustainable economic growth and a high quality of life, with a wise management of natural resources, through participatory governance" (p. 70). Descriptions of smart cities are now appreciating the soft domain aspects like qualities of people and communities as well as ICTs [2,31]. The new perspective that aims to inspire the sense of community among citizens get insights from the previous bottom-up knowledge scheme and recognize the importance of factors that emulates the concept of smart communities where members and institutions work in partnership to transform their environment [5]. Smart communities makes conscious decisions on technology use for tackling societal challenges which results not only in the increase of quality life but also a means to reinventing city's capabilities for new communal practices [16]. The California Institute for Smart Communities could be exemplify among the first to focus on how communities could become smart and how a city could be designed to implement information technologies [1].

The vast range of contexts has led to the formation of a diverse and nebulous smart city design space, where there is little consensus over what smart cities are and what form they should take. This inhibits communal discourse and slows down the development and widespread deployment of smart city technologies and policies [24]. More crucially, it is a barrier to citizen engagement and bottom-up design. Communities are unlikely to engage with, identify, and then design solutions for civic problems while the smart city concept is incoherent, unapproachable, and hard to measure. The agenda for this research is to study the bridge between the soft and hard domain aspects of smart cities and smart communities embedded. On one hand, the hard domain side is where infrastructures such as ICT have a decisive role in the functions of the smart city. On the other hand, the term has also been applied to soft domains where approaches towards culture and social inclusion in a smart city that supposed to offer environments for an entrepreneurship accessible to all citizens. The taken aspect of the smart cities in this research concerns ICT provided opportunities such as social network services and therefore social capital utilization for entrepreneurial oriented activities. Data in social network services as a communication platform will be utilized to study the content and discussions on the innovation and entrepreneurship in on smart city while the general procedure to systematically deal with SNS data will be described. Further, with having the data analyzed and operationalization of the extracted simplified metrics, we attempt to investigate the influential content in SNS regarding the innovation and entrepreneurial discussions. Therefore, the conceptual framework for approaching smart cities within the focus of this research should offer insights regarding the operationalization of social network services data and the effect magnitude of a content in SNS in the context of innovation and entrepreneurship discussions.

3 Innovation and Entrepreneurial Ecosystems and the Role of Social Network Services

Innovation and entrepreneurship concepts are highly intertwined and dependent on each other and are recognized as the core critical components for the wealth and competitiveness of cities and countries [43]. Innovation is an inherently human endeavor, and successful innovation happens when people with skills, experience, and capabilities come together to understand or predict, and then address existing challenges while entrepreneurship is the attempt to setting up and scaling the efforts [15].

Smart cites are introduced as the territories that connects the physical, the IT, the social, and the business infrastructure to leverage the capability of learning and innovation, which is built-in the collective intelligence of the city and its population [23]. The smart infrastructure of cities can tackle the existing challenges in innovation and entrepreneurship ecosystems. In particular, the role of ICT services as one of the dimensions of smart cities can enhance the innovation and entrepreneurship ecosystem. Smart cities have the infrastructure to bridge and facilitate the connectivity of society and in general the social capital for entrepreneurial activity. With the emergence of social network services in the past decade, a new medium has been created to present the society that has not gotten the proper attention yet. The social infrastructure, such as intellectual and social capital, presented by SNSs is an indispensable endowment to the smart cities as it allows, "connecting people and creating relationships" [2]. ICTs also offer new avenues for openness by providing access to social media content and interactions that are created through the social interaction of users via highly accessibly Web-based technologies.

Social media platforms had significant growth over the last decade. According to online statistics and market research source Statista [39], over 70 percent of internet users were social network users in the year 2017 and these figures are expected to grow. It is estimated that the number of social media users will increase from 2.34 billion in 2016 to 2.95 billion in 2020 [39]. Social networking is one of the most popular online activities with high user engagement rates and expanding mobile possibilities. The growth of the SNS's user base is universal and now been increasingly populated and used by much diverse age groups [25]. The growth of social network services is unprecedented that are now so well established and considered a major visited services in internet that doesn't change much from year-to-year [13]. The recent evaluation of actively used social networking services by Pew Internet indicates Facebook as the dominance platform including the owned service of Instagram by 76 percent of active user's login while Twitter is reported to have 42 percent of active user's login [12].

It is therefore reasonable to say that social media represent a revolutionary new trend which have the potential to enhance existing and foster new cultures of openness [6]. Social media empowers its users by the ability to inexpensively publish or broadcast information as it gives them a platform to effectively democratize information and communication real time. Yet, despite the all facilitation of information creation and dissemination, there seems to be very limited understanding of what the "social media" or "social networking services" exactly represent and eventually do to societies. Meanwhile, smart city programs which have received great publicity, there has been less discussion about the evaluation and measurement regimes of societal and soft domain aspects in smart cities. The lack of metric for grasping the societal activities has been depicted in the 'Global Innovators: International Case Studies and Smart Cities' [10] report that notes the inadequacy of existing evaluation approaches which tended to be non-standard, and focused on implementation processes and investment metrics rather than city outcomes and impacts.

This paper aims to investigate the social capital on innovation and entrepreneurship within the smart cities by diving to social networking services as the derivative of one of the major dimensions of smart cities. This research presents utilization of SNSs in understanding and capturing entrepreneurial oriented discussions and further investigates the various profile type impact on SNSs regarding entrepreneurial oriented discussions.

4 Methods

In this section, I share the approach on utilizing computational advancement to analyzing social network services data in a systematic process. The approach uses semantic and linguistics analyses for detecting major topical discussion in the twitter as the SNS platform under study. The following section will describe a general process on SNSs data collection, topic discovery and topic-content analysis. Furthermore, the analysis interpretation discloses insightful characteristics of tweets regarding their topic of discussion and the characteristics of the content generator.

4.1 Systematic Approach to Analyze Social Network Services Data

The data in SNSs often comes unstructured as information that is not organized in a pre-defined manner and not necessarily presents a pre-defined data model. Unstructured information is typically text-heavy, but may contain data such as dates, numbers, and facts as well. Advancements in data mining and text analytics will be obtained in this study to analyses the SNSs data for insightful information.

In this paper, the focus is on getting insight from SNSs as a major component in smart cities regarding entrepreneurial oriented activity. The overall architecture to process data in SNSs is composed and presented graphically in Figure 1. The considered data is collected on Twitter (twitter.com). However, the process has a high extent of generalizability to most of the data in SNSs platforms. The present process included three major phases: capture, curate and consume. In addition, each phase has two subphases consequently according for Figure 1.

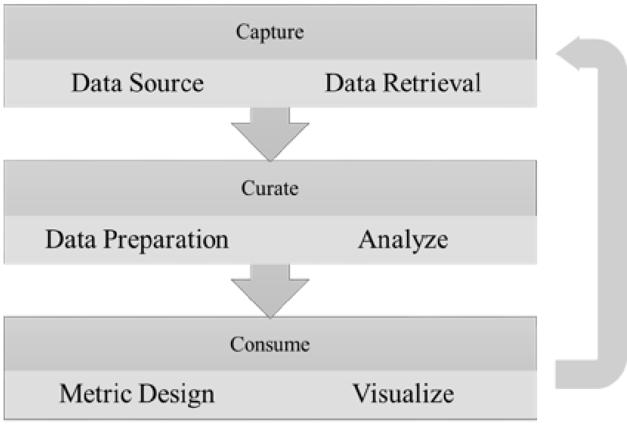


Fig. 1. SNSs systematic data analysis.

Capture: This is the process of collecting data, which contains the selection of the data source, searching for the data and collecting data for other usage. Inputting the searching query is the primary way to specify the content, which is of any interest to retrieve. Various specifications can be implemented, such as keywords, length, date, etc. in order to target the topic of interest. In other words, the required data is obtained by set of criteria embedded with the search query. Some SNSs platforms such as Twitter offer the possibility to retrieve data via the live stream.

Curate: Data curation is a broad term used to indicate processes and activities related to the organization and integration of data collected from various sources. Data retrieval methods are often loosely controlled, resulting in out-of-range values. The data preparation task is performed to reduce the irrelevant and redundant data present in the collected set. This task is necessary for the forthcoming steps so to normalize the data for a better knowledge discovery results. Data analysis can be very subjective to the context of the study and expected results, but the two primary task in analysis can be mentioned as data feature extraction and data classification. The intent for feature extraction is to facilitate the further distinctions and categorization of the data. This task will drive values (features) from the data regarding the context of the knowledge discovery process. Classification of the data occurs in order to reduce the dimensionality of the data. It's an approach derived from the general hypothesis of the knowledge discovery task so to distinguish the best-fit data points from the mass. In this case study, topic modeling has been performed in order to understand the major important cluster of discussions regarding their topics.

Consume: This refers to publishing a presentable format of the information derived from the data. The insights from the results can be provided in visually appealing way or can be used as a metric to be combined with other data points for further interpretations. Having the systematic social network services data analysis explained, the next section, the presented procedure will be applied on a case study.

4.2 Evaluating Entrepreneurial Oriented Activity in Twitter : London City Case Experiment

The background literature discusses the importance of emerging social network services in smart cities and the need for investigating the effect of entrepreneurial discussions in innovation ecosystem. In this section, we utilize the systematic approach on analyzing SNSs data and emphasize on the new ways of benchmarking for social capital by focusing on social network services. In order to solidify the objective, an experiment has been condicted so to detect and capture entrepreneurial discussions on one of the dominant social network services called Twitter. A popular microblogging tool Twitter, has seen a lot of growth since it was launched in October 2006; is an online news and social networking service where users post and interact with messages called ”tweets”, restricted to 140 characters. Twitter users can post their opinions or share information about a subject to the public. Twitter has 316 million users worldwide [14], providing a unique opportunity to understand societal discussions and in this study case a way to comprehend entrepreneurial oriented discussion.

The initial interest of the study was to capture innovation and entrepreneurial oriented discussion from social network services as one of the major themes that needs studying in smart cities. Start-ups are considered as a good representation of societal practice of entrepreneurship. Start-ups are increasingly seen as significant contributors to national job-creation [38]; employment and gross national product data demonstrated the shift to an innovative start-up dominated economy [38]. Therefore, fostering the start-up ecosystem is seen as the measure for improving national economy [35]. The study case experiment has been conducted to collect the activity related to the start-up ecosystem in the studied country so to capture the relevant societal discussions oriented towards innovation and entrepreneurship.

Twitter is an SNS platform, which well represents and acts as support infrastructure for start-ups, which organically are socially active. The study took the initiative to collect a sample of tweets from a region (country) and extract features (words and hashtags) related to start-up activities; we have applied techniques to decompose hashtags, analyze them, and reuse the information extracted for classification purposes. Twitter provides application programming interface (APIs) to access tweets and information about posted content and users. The potential bias of Twitter APIs was discussed by a recent research [20]. Twitter data has been used for a wide range of studies such as stock market [8], brand analysis [22] and election analysis [41]. The unique characteristics and features of Twitter as a microblogging service are illustrated in Figure 2.

Post	Body	Urls
		Usermention
		Tween Lang
		Media (Video, Picture)
		Hashtag
	Provider	FollowesCount
		Follwed by Count
		Profile Description
	Location	Country/City /State
	Like counts	
	Link	
	Retweet count	
	Posted time	

Fig. 2. Twitter Meta data illustration.

With respect to Twitter's characteristics, a multi-component semantic and linguistic framework was developed to collect Twitter data, prepare and analyze the data and discover insightful information. In order to demonstrate the steps for utilizing SNSs data for valuable insights, a high ranked smart city has been selected. London considered as one of the top smart city in global scale [18,21] and as the English is the dominant language; this will facilitates the text analytics tasks. With respect to Twitter's characteristics, the search queries were constructed in a way that captures the most relevant content regarding start-up scene and the entrepreneurial activity.

4.3 Data collection (Capture)

This phase attempt was to collect relevant tweets using Twitter's Application Programming Interfaces (API) [42]. We have benefited from popular hashtag recommender toolkits such as http://hashtagify.me, "https://ritetag.com" and "https://www.trendsmap.com" to discover the relevant hashtags and their proximities to the innovation and entrepreneurial oriented discussions. Figure 3 is illustrating the hashtags proximity with the subject of our initial search (#startup #startups #entrepreneur #tech #sme #innovation #entrepreneurship #startuplife # hackathon) which obtained for detecting the extended hashtags and relevant discussions.

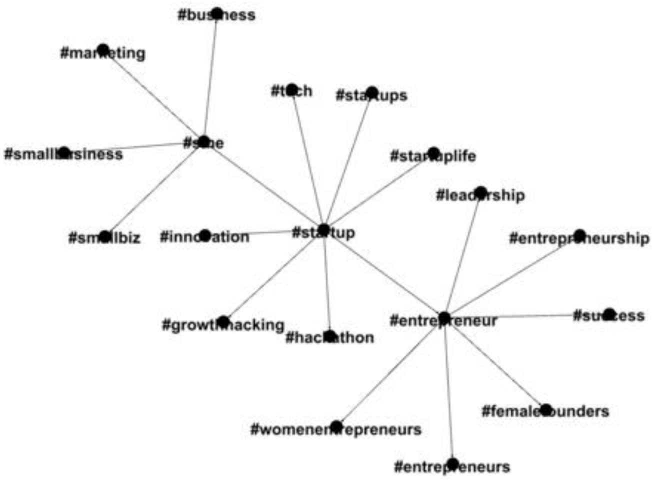


Fig. 3. Twitter hashtag proximity distance.

Twitter's API provides both historic and real-time data collections. The latter method randomly collects 1 percent of publicly available tweets. We used the real-time method to randomly collect 10 percent of publicly available English tweets using several predefined hashtags related queries mentioned previously within a specific period. We used the extended query to collect approximately 4 thousand related tweets between 06/01/2017 and 08/30/2017. The data will be available in the following link “https://goo.gl/mZumDp”. Table 2 shows a sample of collected tweets textual content, users and overall interaction (sum of likes and retweets) for each tweet in this research.

4.4 Curate

This phase, the analysis of tweets by data feature extraction and data classification has been advanced. Regarding the SNSs data which is collected from twitter. The investigations began with an empirical analysis of the dynamics of the discussions in the Twitter. The topical structure of discussions has been studied. Further, we will investigate the characteristics of the major content producers. The Twitter analytic process was facilitated by Azure cloud computing platform (azure.microsoft.com) which the pipeline of the process can be seen in Figure 4.

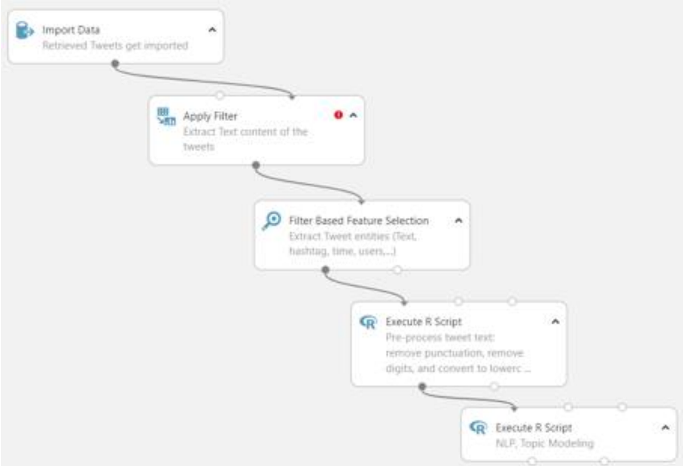


Fig. 4. Twitter content analysis with Azure Cloud Computing Platform.

After importing the retrieved tweets as the input data, a filtering process applies to structure and reduce the noise of the data. The data feature extraction distinguishes the valuable data points such as number of retweets, likes, profile identifications and the textual content of the tweets as we will leverage these data point for further insights. One classification task for analyzing tweets; topic modeling has been utilized in order to reveal the topical formation of the discussion. Topic modelling can be described as a method for finding a group of words (i.e. topic) from a collection of documents (in our case tweets) that best represents the information in the collection. It can also be thought of as a form of text mining – a way to obtain recurring patterns of words in textual material [37]. There are many techniques that are used to obtain topic models in this study we leveraged Latent Dirichlet Allocation (LDA) and the consequent visualization toolkit developed for that (LDAviz) so to visually show the major twitter discussion topics [7]. The next section we will represent the classification calculation results visually.

4.5 Results (consume)

So far, we were able to encapsulate the entrepreneurial oriented activity via focusing on start-up scene in the smart city of London. The dynamic relevant discussions in social network services (in this study Twitter) were captured and curated to transform the SNSs data into insightful information. The dynamic discussions and interactions on SNSs regarding entrepreneurial oriented matters can represent the social capital as explained in earlier sections. In this section, we will dive deeper into SNSs data in order to detect the most influential content and type of content generator profiles associated. A categorization analysis task will be performed into the textual content of the SNSs data in order to get a broad overview and distinguish the general topic of discussions.

The analysis of topical structure of SNSs discussion with LDA is visualized in Figure 5, which illustrates the general topical theme of discussions. The six major clusters are named based on the major keywords mentioned under each topic. The visualization also revels the size of the discussion proportional to other topics via their circle size and indicates the distance of topics in two dimensional distance map.

Fig. 5. Intertopic Distance Map.

As part of data consumption and insight generation task, with having the meta data of each posted tweet and the associated profile under each of the topics, the influential profiles based on their overall interaction (Number of retweet and likes received for the post) can be detected. This information will reveals how contents (tweets) gets attentions in different topics regarding their content generators. The motivation for content generators in twitter profile categorization stems largely from the fact that humans as intelligent individuals impose complex factors on the consumption and dissemination of information on SNSs [26,29]. Therefore, as the different profile types have different purposes and cater to different needs, the categorization of content generators in each of the six topical discussions will help us to measure the impact and influence each category is making. The categorization definitions and process inspired from Uddin et al. [43] and due to the study intentions, three major different types of Twitter profile defined and were developed which are as follows:

Personal profiles: These accounts contain personal content, have no ties to business, and do not mention corporate or brand information. They are created by individuals who do not wish to be identified with their employer. Technically, the accounts have been created to acquire news, learn, have fun, etc. Generally, these individuals exhibit low to mild behaviour in their social interaction. *Professional profiles:* Personal users who communicate their professional views on Twitter. They share useful information on specific topics and are involved in healthy discussion related to their specialist interests and expertise. Professional users tend to be highly interactive; they follow many and are followed by many. *Corporate and business profiles:* Different to personal and professional users in that they follow a marketing and business agenda on Twitter. Their profile description accurately describes their motives, and similar behaviour can be observed in their tweeting patterns. Frequent tweeting and less interaction are the two key factors that separate business users from both personal and professional users. The type of content will be primarily corporate. Such accounts are often managed by company teams working under a specific brand name related to the company, providing corporate news and support.

Under each of the six discussion topics, profiles ranked based on their tweet interaction ratio (number of retweets + number of likes) were manually looked and categorized according to the three major profile descriptions. Figure 6 is an illustration of the manual categorization of the top content generator profiles.

Interaction by count of Retweets	Personal profiles	Professional profiles	Corporate and business profiles
Educational	35	96	52
Motivational	40	65	18
Promotion	10	62	31
News	5	26	56
Events	12	31	24
Viral/Marketing	0	18	17
Interaction by count of Likes	**Personal profiles**	**Professional profiles**	**Corporate and business profiles**
Educational	24	187	37
Motivational	18	275	87
Promotion	14	33	12
News	44	87	125
Events	65	77	42
Viral/Marketing	0	18	25

Fig. 6. Categorization of tweets based on topic and generator.

As it can be observed from Figure 6, professional users have more influence in overall. In topical content categories, professional users are generating the highest influence in educational, motivational, promotion and events type of topics. Corporate and business profiles tend to be more influential in news category, educational, and promotional after professional users. Counting the likes, the calculation reveals that professional users have more interaction, especially in educational and motivational content category, while business profiles have the higher interaction in the news category and motivational category in second order. Personal profiles have the lowest influence among the other two profile categories in both retweets and count of likes. The difference in distribution of interaction is that motivational and educational receives the highest retweets and in the calculation of like counts, the high-interacted categories will shift to events and news.

References

1. Suha Alawadhi, Armando Aldama-Nalda, Hafedh Chourabi, et al. 2012. Building understanding of smart city initiatives. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 7443 LNCS: 40–53. http://doi.org/10.1007/978-3-642-33489-4_4

2. Vito Albino, Umberto Berardi, and Rosa Maria Dangelico. 2015. Smart Cities: Definitions, Dimensions, Performance, and Initiatives. Journal of Urban Technology 22, 1: 3–21. http://doi.org/10.1080/10630732.2014.942092
3. Ari Veikko Anttiroiko. 2016. City-as-a-platform: The rise of participatory innovation platforms in finnish cities. Sustainability (Switzerland) 8, 9. http://doi.org/10.3390/su8090922
4. Alan Beattie. 2009. False Economy: A Surprising Economic History of the World. http://doi.org/10.1038/480005b
5. U. Berardi. 2013. Sustainability Assessments of urban Communities through Rating Systems. Environment, Development and Sustainability 15, 6: 1573–1591.
6. John C. Bertot, Paul T. Jaeger, and Justin M. Grimes. 2010. Using ICTs to create a culture of transparency: E-government and social media as openness and anti-corruption tools for societies. Government Information Quarterly 27, 3: 264–271. http://doi.org/10.1016/j.giq.2010.03.001
7. David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." Journal of machine Learning research 3.Jan (2003): 993-1022.
8. Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. Journal of Computational Science 2, 1: 1–8. http://doi.org/10.1016/j.jocs.2010.12.007
9. Volker Buscher and Léan Doody. 2013. Global Innovators: International Case Studies on Smart Cities.
10. Sally Caird, Lorraine Hudson, and Gerd Kortuem. 2016. A Tale of Evaluation and Reporting in UK Smart Cities, Open Research Online.
11. Andrea Caragliu, Chiara del Bo, and Peter Nijkamp. 2011. Smart cities in Europe. Journal of Urban Technology 18, 2: 65–82. http://doi.org/10.1080/10630732.2011.601117
12. Pew Research Center. 2017. Social Media Fact Sheet. Retrieved October 1, 2017 from http://www.pewinternet.org/fact-sheet/social-media/
13. Dave Chaffey. 2017. Global social media research summary 2017. smartinsights. Retrieved October 2, 2017 from http://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/
14. D. Olanoff. 2015. Twitter Monthly Active Users Crawl To 316M, Dorsey: We are not satisfied.
15. Peter Drucker. 2012. Innovation and entrepreneurship. Routledge.
16. John M Eger. 2009. Smart Growth, Smart Cities, and the Crisis at the Pump A Worldwide Phenomenon. I-Ways 32, 1: 47–53. http://doi.org/10.3233/IWA-2009-0164
17. Johannes C. Eichstaedt, Hansen Andrew Schwartz, Margaret L. Kern, et al. 2015. Psychological Language on Twitter Predicts County-Level Heart Disease Mortality. Psychological Science 26, 2.
18. Forbes. 2017. Forbes. Retrieved from https://www.forbes.com/sites/iese/2017/05/31/the-smartest-cities-in-the-world-for-2017
19. Gartner. 2011. Hype Cycle for Smart City Technologies and Solutions. Retrieved from https://www.gartner.com/doc/1754915/hype-cycle-smart-city-technologies
20. Sandra González-Bailón, Ning Wang, Alejandro Rivero, Javier Borge-Holthoefer, and Yamir Moreno. 2014. Assessing the bias in samples of large online networks. Social Networks 38, 1: 16–27. http://doi.org/10.1016/j.socnet.2014.01.004
21. Alex Gray. 2017. These are the best-run cities in the world. World Economic Forum. Retrieved from https://www.weforum.org/agenda/2017/10/worlds-best-run-cities-jll/
22. Arash Hajikhani, Jari Porras, and Helinä Melkas. 2016. Brand Analysis in Social Network Services: Results from Content Analysis in Twitter Regarding the US Smartphone Market. International Journal of Innovation and Technology Management: 1740008. http://doi.org/10.1142/S0219877017400089

23. C. Harrison, B. Eckman, R. Hamilton, et al. 2010. Foundations for Smarter Cities. IBM Journal of Research and Development 54, 4: 1–16.
24. Robert G. Hollands. 2008. Will the real smart city please stand up? City 12, 3: 303–320. http://doi.org/10.1080/13604810802479126
25. Andreas M. Kaplan and Michael Haenlein. 2010. Users of the world, unite! The challenges and opportunities of Social Media. Business Horizons 53, 1: 59–68. http://doi.org/10.1016/j.bushor.2009.09.003
26. Laeeq M Khan. 2017. Social media engagement - What motivates user participation and.pdf. Computer in Human Behavior 66: 236–247.
27. Karima Kourtit and Peter Nijkamp. 2013. In praise of megacities in a global world. Regional Science Policy & Practice 5, 2: 167–182. http://doi.org/10.1111/rsp3.12002
28. Patrizia Lombardi, Silvia Giordano, Hend Farouh, and Wael Yousef. 2012. Modelling the smart city performance. Innovation 25, 2: 137–149. http://doi.org/10.1080/13511610.2012.660325
29. JT Lussier, T Raeder, and NV Chawla. 2010. User Generated Content Consumption and Social Networking in Knowledge-Sharing OSNs. International Conference on Social Computing, Behavior Modeling and Prediction.
30. M. Merli and E. Bonollo. 2014. Performance Measurement in the Smart Cities. Smart City : how to create public and economic value with high technology in urban space: 139–155.
31. T. Nam and T.A. Pardo. 2011. Smart city as urban innovation: Focusing on management, policy, and context. ACM International Conference Proceeding Series. http://doi.org/10.1145/2072069.2072100
32. Paolo Neirotti, Alberto De Marco, Anna Corinna Cagliano, Giulio Mangano, and Francesco Scorrano. 2014. Current trends in smart city initiatives: Some stylised facts. http://doi.org/10.1016/j.cities.2013.12.010
33. Peter Nijkamp and Karima Kourtit. 2013. The "New Urban Europe": Global Challenges and Local Responses in the Urban Century. European Planning Studies 21, 3: 291–315. http://doi.org/10.1080/09654313.2012.716243
34. Chris Richter, Sascha Kraus, and Pasi Syrjä. 2015. The Smart City as an opportunity for entrepreneurship. International Journal of Entrepreneurial Venturing 7, 3: 211. http://doi.org/10.1504/IJEV.2015.071481
35. Renata Lebre La Rovere, Luiz de Magalhaes Ozorio, and Leonardo de Jesus Melo eds. 2015. Entrepreneurship in BRICS: Policy and Research to Support Entrepreneurs. ix.
36. Jesse Shapiro. 2005. Smart Cities: Quality of Life, Productivity, and the Growth Effects of Human Capital. Cambridge, MA. http://doi.org/10.3386/w11615
37. Carson Sievert and Kenneth Shirley. 2014. LDAvis: A method for visualizing and interpreting topics. Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces: 63–70.
38. J E Sohl. 2006. Angel investing: Changing strategies during volatile times. Journal of Entrepreneurial Finance and Business Ventures 11, 2: 27–47. Retrieved from http://www.econstor.eu/bitstream/10419/55934/1/662802578.pdf
39. Statista. 2017. Number of social network users worldwide from 2010- 2018. Statista. Retrieved from https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/
40. Moe Thuzar. 2012. Urbanization in Southeast Asia: Developing SMart Cities for the Future? Regional Economic Outlook: 96–100.
41. Andranik Tumasjan, To Sprenger, Pg Sandner, and Im Welpe. 2010. Predicting elections with Twitter: What 140 characters reveal about political sentiment. Proceedings of the

Fourth International AAAI Conference on Weblogs and Social Media: 178–185. http://doi.org/10.1074/jbc.M501708200
42. Twitter. 2017. Twitter developer documentation. Retrieved from https://dev.twitter.com/docs
43. Muhammad Moeen Uddin, Muhammad Imran, and Hassan Sajjad. 2014. Understanding Types of Users on Twitter. 6. Retrieved from http://arxiv.org/abs/1406.1335
44. UNPD. 2007. World Urbanization Prospects: The 2007 Revision Population Database. Retrieved from http://esa.un.org/unup/
45. Janyce Wiebe, Eric Breck, Chris Buckley, and Claire Cardie. 2003. Recognizing and Organizing Opinions Expressed in the World Press. In Working Notes - New Directions in Question Answering (AAAI Spring Symposium Series: 12–19. Retrieved from http://www.aaai.org/Papers/Symposia/Spring/2003/SS-03-07/SS03-07-003.pdf
46. World Economic Forum. 2013. Entrepreneurial Ecosystems Around the Globe and Company Growth Dynamics. Report Summary for the Annual Meeting of the New Champions 2013, September: 36. Retrieved from http://www3.weforum.org/docs/WEF_EntrepreneurialEcosystems_Report_2013.pdf
47. World Economic Forum. 2014. The Competitiveness of Cities: A report of the Global Agenda Council on Competitiveness. Retrieved from http://www3.weforum.org/docs/GAC/2014/WEF_GAC_CompetitivenessOfCities_Report_2014.pdf
48. J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack. Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques. Third IEEE International Conference on Data Mining: 427–434.

Collective consciousness in business ecosystems

Marja Turunen [1, 2] and Matti Mäntymäki [1]

[1] Turku School of Economics, University of Turku, Finland
[2] School of Science, Aalto University, Espoo, Finland
marja.turunen@utu.fi, matti.mantymaki@utu.fi

Abstract. This paper presents collective consciousness as a lens through which to analyze the psycho-social dynamics of business ecosystems. While the business ecosystem concept has drawn a lot of attention in software and business literature, the intangible psycho-social layers of attention and shared cognition produced by the interactions between ecosystem actors are not well understood. To address this void in the literature, we adopt collective consciousness as a conceptual tool to better understand business ecosystems as complex networks of heterogeneous actors. We present an illustrative case of an emerging business ecosystem of digital services for real estate and facility management and scrutinize the applicability of collective consciousness as a conceptual device to better understand the characteristics and dynamics of business ecosystems. We suggest that employing collective consciousness provides a useful analytical device to better understand the complexities emerging from the interactions between the actors. We further discuss under what circumstances employing collective consciousness as a conceptual tool adds particular value for business ecosystem research and practice.

Keywords: business ecosystem, collective consciousness, digitalization, conceptual analysis, digital transformation, digital disruption

1 Introduction

This paper presents collective consciousness as a lens through which to analyze the psycho-social dynamics of business ecosystems. The term *ecosystem* has been widely adopted in the business and technology literature as a metaphor to describe certain types of business networks (e.g., Autio & Thomas, 2014; Hyrynsalmi, 2015; Mäntymäki & Salmela, 2017; Hyrynsalmi, Mäntymäki, & Baur, 2017; Teece, 2010; Mäntymäki, Salmela, & Turunen, 2018).

The current business and technology literature includes a number of variants of the ecosystem concept, such as business ecosystems (Peltoniemi & Vuori, 2004), innovation ecosystems (Oh, Phillips, Park, & Lee, 2016), software ecosystems (Hyrynsalmi, Suominen & Mäntymäki, 2016), service ecosystems (Vargo & Lusch, 2010), product ecosystems (Frels, Shervani, & Srivastava, 2003), and platform ecosystems (Ceccagnoli, Forman, Huang, & Wu, 2013), to name but a few. At the same time, however, the use of the ecosystem metaphor has also been criticized, and the accuracy of the metaphor has been questioned (Oh, Phillips, Park, & Lee, 2016; Hyrynsalmi, 2015; Hyrynsalmi & Mäntymäki, 2018).

The term ecosystem emanates from ecology, where it typically denotes a unit of biological organization made up of all the organisms in a given area, thus forming a "community." Organisms within a community interact with the physical environment so that the flow of energy leads to a characteristic trophic structure and material cycles within the system (Odum, 1966).

The concept of collective consciousness in turn emanates from the social sciences, particularly social psychology and sociology (Vygotsky, 1980; Hutchins, 1995), and originally dates back to Durkheim (1895). In his studies of the sociology of suicide, Durkheim (1951) found out that individuals' acts, such as suicide, depended on the collective consciousness within a society. Thereafter, collective consciousness has been examined in a wide range of contexts, including business networks (Allee, 2003; Normann & Ramirez, 1993; Normann, 2001) and business ethics (Pandey & Gupta, 2008).

Interestingly, however, the intangible psycho-social dimensions and the associated complexity of business ecosystems have thus far received less scholarly attention. To address this void in the current body of knowledge, we employ the concept of collective consciousness to scrutinize the intangible elements of business ecosystems. In doing so, we follow Tsoukas (2017), who maintained that increasing the complexity of organizational theory is essential to better capture the complex nature of real-life organizational phenomena. To this end, we adopt Turunen's (2015) view of organizing, which maintains that the ecosystem conceptualization is embedded in organizational consciousness.

The purpose of this paper is to understand if and how collective consciousness manifests itself in business ecosystems. We present an illustrative case study of an emerging business ecosystem for internet of things (IoT)–driven real-estate and facility services. This study contributes to the business ecosystem literature by delineating a need for increased analytical depth and conceptual clarity in studying the intangible elements and dynamics of business ecosystems. We further conclude that additional scrutiny of the ecosystem metaphor and the value it adds to theorizing and managerial communication is needed.

The paper proceeds as follows: After the introductory section, we present a discussion of the business ecosystem concept. Thereafter, we present a set of related constructs used to depict business networks and analyze how they converge with, and diverge from, the business ecosystem construct. The paper concludes with a synthesis of the analysis, a reflection upon an emerging digital business ecosystem in real-estate and facility management, and lastly presents suggestions for the future research.

2 A consciousness-based view of business ecosystems

2.1 Collective consciousness

In sociology, the term *collective consciousness* dates back to Durkheim (1895). Durkheim depicted collective consciousness as an awareness of something bigger than the individual, such as the shared understanding of social norms, and those norms are able to affect individuals. In sociology and social psychology, collective consciousness (Vygostky, 1978; Leontjev, 1973; Bronfenbrenner, 1977) has been viewed to manifest itself in people's activities in the world, particularly in symbolic systems such as language. This perspective is exemplified by Leontjev (1973, p. 183), who viewed consciousness as pervading all human actions, with activity being an important substance of consciousness. As a result, collective consciousness is distinct from individual-level consciousness (Turunen, 2014, 2015).

According to Vygotsky (1980), collective consciousness is a purpose- and meaning-making dimension above any individual actor (individuals, organizations, or society). While actors employ their individual consciousness, collective consciousness emerges and is constituted through interactions and meaning-making between actors (Turunen, 2015). The interactions between actors generate relational consciousness in the collective field. In addition, actors interact with other relational fields beyond their ecosystem. Second, actors engage in the meaning-making of their own entity, such as their business, group or organization. In the previously described interactions, relational meaning-making is built. Furthermore, part of the meaning-making is involved with other actors beyond the ecosystems of an actor.

The borders of collective consciousness are an inevitable dynamic. An individual actor—that is, an individual, a group, or an organization—can access the collective consciousness. However, an individual actor cannot reach the whole picture and totality of collective consciousness (Vygotsky, 1980; Hutchins, 1993). At the same time, an individual is influenced by the collective consciousness, often through subconscious processes, habits, and routines.

Since collective consciousness is essentially socially constructed, it is affected by, and has an impact on, a number of individual-, group-, and society-level contingences, such as trust, norms, and values, to name but a few. Because collective consciousness is based upon reciprocal ties between actors, it may play a focal role in inducing transformation and renewal but also repression and stagnation. In any case, the process of developing the consciousness and the artifacts produced by the process (Garud & Turunen, 2014, 2017) need simultaneous attention. In the next subsection, we discuss the business ecosystem concept from a collective-consciousness perspective.

2.2 Business ecosystem

The business ecosystem concept was developed by Moore (1993). His seminal article describes capability coevolution with innovation with distinct stages towards a shared future and an accruable profit model of the business ecosystem. Recently, Mäntymäki, Salmela, and Turunen (2018) found that business ecosystems appear to have three characteristic features: First, members of an ecosystem are highly interconnected. Interconnectedness refers to the fact that the success or failure of a member of an ecosystem affects the other members. Second, a business ecosystem often includes a keystone that "regulates ecosystem health" (Moore, 1993, p. 8). The keystone is typically an actor that is able to support and orchestrate the activities that take place within the ecosystem. Third, ecosystems are complex systems (Peltoniemi & Vuori, 2004). As described by Cowan (1994, p. 1), complex systems "contain many relatively independent parts which are highly interconnected and interactive." Lewin (1999) in turn further contends that complex systems are systems whose properties are not fully explained by an understanding of its constituent parts. Thus, complex systems can be informed by the research stream of process studies (James, 1977; Tsoukas & Chia, 2002) of collective interaction (Kimble, 2008).

While explicit notions of collective consciousness are missing in the ecosystem concept in explicit terms, collective consciousness is accommodated in ecosystem terms most clearly in the value network. In fact, collective consciousness is fostered by the interrelations between individuals, groups, and organizations. The contributors to the concept of value networks mentioned the benefits of collective consciousness explicitly, such as Normann (2001) and Allee (2003). For instance, Allee (2003, p. 54) maintained that "collective consciousness provides a new transformative shift towards understanding the more complex layers of the system and new avenues for connecting together with other players"—that is, collaboration in the intangible areas of value creation.

Mäntymäki et al. (2018) explained that the business ecosystem concept contains an internal tension. The current consensus presupposes that a business ecosystem is a collective entity that is regulated and/or orchestrated by a single dominant actor. However, a deeper examination of the social dimensions of business ecosystems implies that ecosystem actors may over time develop a common awareness of the ecosystem entity that helps them to manage and make sense of the diversity and complexity of the network. Against this backdrop, we discuss how collective consciousness manifests itself in the key criteria Mäntymäki and Salmela (2017) used to evaluate different types of business networks, including the definition of group borders, the nature of relationships between actors, sources of transformation and change, and applicability. Table 1 presents the dimensions of consciousness and their descriptions for an emerging ecosystem.

Table 1. Dimensions of the collective consciousness for an emerging ecosystem

Dimension	**Description**
Definition of group borders	Collective consciousness is constructed in a web of actions and relationships that generates both tangible and intangible value through complex dynamic exchanges between two or more individuals, groups, or organizations.
Primary relationship between actors	Collective consciousness is an intangible, connected field available for each actor that enables a connection to the larger system.
Sources of transformation and change	Collective consciousness provides a new transformative shift towards understanding the more complex layers of the system and new avenues for connecting and exchanging information together with other players.
Applicability	Collective consciousness can explain the reasons how ecosystems may flourish and be able to generate big leaps, enable strategic collaboration, and information exchange between diverse organizations and individuals with partly shared and competitive/diverse motives.

3 The case of an emerging business ecosystem for digital real-estate and facility services

Advances in digital technologies, for example, in sensor technology and IoT (cf. Mian, Mäntymäki, Riekki, & Oinas-Kukkonen, 2016), fuel the generation of data (Koskenvoima & Mäntymäki, 2015) and thus enable the creation of new value networks and business models (cf. Wirén & Mäntymäki, 2018; Xu, Turunen, Ahokangas, Mäntymäki, & Heikkilä, 2018) for established, mature businesses.

This in turn often challenges the existing logic of value creation. For example, in the digital real-estate and facility business ecosystem, the collective consciousness of a value network can be viewed as being interwoven into the value-creation process. As a consequence, the value constellation created by the ecosystem crystallizes and may start to appeal to new actors, who join in the value creation and affect the contextual dimensions of the ecosystem (Xu et al., 2018).

We illustrate this process with a case of an emerging business ecosystem for IoT-driven real-estate and facility services. The research and development activities toward the creation of a new business ecosystem are supported by Business Finland, a key source of public research and development funding in Finland. The purpose of the ecosystem initiative was to ignite a set of activities to develop new end-user services for the real-estate and facilities business by leveraging IoT, sensor technology, and artificial intelligence. The tangible activities within the initiative have been divided into four thematic entities, titled well-being, intelligent restaurant, data-as-services, and empathetic building.

We start our analysis by identifying the different actor types involved in the ecosystem (cf. Islam, Mäntymäki & Turunen, 2019) and scrutinizing their potential influence on the collective consciousness in the network. Table 2 provides a summary of the analysis.

Table 2. The identified actor types influencing the qualities of collective consciousness in the emerging business ecosystem

Actor type	**Critical dimensions**	**Description**	**Collective consciousness illustrative outcomes**
Individual participant or ecosystem representative.	Individual consciousness, reference to cultural base, knowledge, value, meaning-making, interrelations, and digital and social media presence.	Individual actors, such as an individual or organization, with a single participant in the project/ecosystem.	Dependent on the intensity of the interactions and qualities of the individual with others.
Organization accommodating participants and organizations and institutions fostering ecosystem development.	Organizational consciousness and culture, such as strategic intentions, value system, distribution of information and power, communication channels, artificial intelligence, and the digitalization phase.	The dynamics of a value networks are visible to a certain degree. Participants collaborate in and negotiate the value constellation. Interactions in the value network create collective consciousness.	A mixture of the individual and collective consciousness.
Project organization.	Intervened by the consciousness of the dominant players in the ecosystem and the aggregated project consciousness.	The project consciousness is not a direct aggregation of the project participants' qualities of consciousness. The dominant roles in the project, such as project leader, affect every participant by their consciousnesses.	A loose aggregate of individual and collective consciousness. The project organization has power over the collective consciousness development.
Emerging ecosystem, a complex system.	Depending on the fit of the competencies and interactions with the other participants and their own reference group, a whole system transformation is possible during the ecosystem evolution.	Provides an alternative explanation of the value of the interaction and information exchange between the ecosystem players. Points out the importance of the qualities of the interaction, such as trust and shared values, in a digital platform.	A new collective cultural layer supporting the ecosystem or, in the worst case, a collective consciousness holding back and preventing the full potential of the ecosystem outcomes.

In our analysis, we viewed the actor type, such as an individual, organization, project, or emerging ecosystem, as pertaining to a particular constellation of collective consciousness, including ties to the collective consciousness of other actors and the intensity of the interaction. The illustration of the possible outcomes of collective consciousness in turn indicates, for instance, opportunities to influence the critical dimensions of the collective consciousness. Consequently, each actor of the ecosystem influences the quality of the collective consciousness. Furthermore, collective consciousness is contingent upon the intensity, frequency, and quality of the interactions between the actors. As a result, in its current state, the emerging ecosystem appears to resemble what the literature refers to as a value network (Allee, 2003). This is due to the fact that the value network concept does not assume the existence or emergence of a dominant player. However, in our case, it is possible that some of the actors make a deliberate effort to take a dominator role in the ecosystem and, thus, in the production of collective consciousness.

The lack of a clear dominator may, on the one hand, increase the need for additional negotiation and thus slow down the development activities. On the other hand, it also may force the actors to articulate their needs and intentions and take a greater responsibility in the overall course of action.

4 Discussion

This study set out to understand if and how collective consciousness manifests itself in business ecosystems. To this end, we presented an illustrative case study of an emerging business ecosystem for IoT-driven real-estate and facility services. We highlight three main findings from the study.

First, collective consciousness appears to provide a conceptual tool to describe and examine how the actors of a business ecosystem deal with the complexities and uncertainties inherent to a networked mode of operation. Hence, our study adds to those by Allee (2003) and Normann (2001), who employed collective consciousness to study value networks, and Hutchins (1995), who highlighted collective cognition.

Second, we conclude that the concept of business ecosystem appears to enable the analysis of both collaborative and competitive relationships. In this regard, the business ecosystem diverges from other concepts used to describe business networks (cf. Mäntymäki & Salmela, 2017; Mäntymäki et al., 2018). These collaborative and competitive interactions in turn may result in unique properties in terms of how they generate collective consciousness.

Third, we point out collective consciousness may be beneficial in dealing with the complexity pertinent to dynamic multi-actor networks such as business ecosystems. Using theoretical and conceptual tools that can explain the research problem with minimal complexity is generally considered a virtue in research. At the same time, however, overly simplistic theoretical and conceptual tools may not be sufficient to identify solutions for highly complex problems (Boulding, 1956). For example, inter-organizational collaboration generates different levels and qualities of attention (Ocasio, 1997; Teece, 2007), such as collective awareness and, consequently, collective consciousness. This in turn can help in dealing with complex issues and problems, including innovations, sustainability, and ethics (Turunen, 2015, 2018; Garud, Turunen & Karunakaran, 2018 a,b).

We conclude that collective consciousness may produce certain intellectual assets for describing and explaining a transformative change that takes place within a complex system. We further argue that this transformative change is a key attribute and characteristic of a business ecosystem.

Like any other piece of research, this study suffers from a number of limitations. First, the study was of a conceptual nature. Future research could seek to empirically examine how collective consciousness may manifest itself in the context of business ecosystems. For example, investigating the nature of relationships between collective consciousness and trust in business ecosystems would potentially significantly add to the current knowledge of ecosystem dynamics (cf. Basole, Russell, Huhtamäki,, Rubens, Still, & Park, 2015; Mäntymäki, 2008).

Second, in addition to the business ecosystem, the literature contains a number of other concepts used to describe business networks (Mäntymäki & Salmela, 2017). Future research should thus incorporate, for instance, platforms and alliances in the analysis. However, there are presumably different types of business ecosystems. Future studies could thus identify different types of business ecosystems and examine if and how they differ in terms of collective consciousness.

REFERENCES

Allee, V. (2003). The future of knowledge: Increasing prosperity through value networks. Routledge.

Autio, E., & Thomas, L. (2014). Innovation ecosystems. The Oxford handbook of innovation management, 204-288.

Basole, R. C., Russell, M. G., Huhtamäki, J., Rubens, N., Still, K., & Park, H. (2015). Understanding business ecosystem dynamics: A data-driven approach. ACM Transactions on Management Information Systems (TMIS), 6(2), article no. 6.

Bengtsson, M., & Kock, S. (2000). Coopetition" in Business Networks—to Cooperate and Compete Simultaneously. Industrial Marketing Management, 29(5), 411–426.

Boulding, K. E. (1956). General system theory – The skeleton of science. Management Science, 2(3), 197-208.

Bronfenbrenner, U. (1977). Toward an Experimental Ecology of Human Development. American Psychologist, 32(7), 500-513.

Ceccagnoli, M., Forman, C., Huang, P., & Wu, D. J. (2012). Cocreation of Value in a Platform Ecosystem! The Case of Enterprise Software. MIS Quarterly, 263-290.

Durkheim, E. (1951). Suicide: A study in sociology.)Glencoe, IL: Free Press.(Original work published 1897).

Frels, J. K., Shervani, T., & Srivastava, R. K. (2003). The integrated networks model: Explaining resource allocations in network markets. Journal of marketing, 67(1), 29-45.

Garud, R. & Turunen, M. (2014). Harnessing Ambiguity for Innovation. Academy of Management Proceedings, vol. (1) pp. 10801 doi: 10.5465/AMBPP.

Garud, R. & Turunen, M. (2017). The Banality of Organizational Innovations: Embracing the Substance-Process Duality. Innovation: Organization & Management, 19, (1), 31-38.

Garud, R., Turunen, M. & Karunakaran, A. (2018a). Organizing For Serendipity At Work. In the Sub-theme 34: Organized Creativity: Harnessing Serendipity and Surprise of the 34th EGOS Colloquium in Tallinn, Estonia, July 5-7, 2018.

Garud, R., Turunen, M. & Karunakaran, A. (2018b). Organizing for Ongoing Innovation at Organizations: A Narrative-Infused Design Approach. The Symposium 'Can large firms really leverage radical innovation? In search of new dimensions and strategies. In the Academy of Management Proceedings, vol. (1) pp.

Hutchins, E. (1995). Cognition in the Wild. Cambridge, MA: MIT press.

Hyrynsalmi, S. (2015). Letters from the war of ecosystems (doctoral dissertation). University of Turku, Finland.

Hyrynsalmi, S. Mäntymäki, M. & Baur, A. (2017). Multi-homing and Software Firm Performance. In Proceedings of the 16th IFIP Conference on E-Business, E-Services and E-Society (I3E2018). Lecture Notes in Computer Science.

Hyrynsalmi, M. & Mäntymäki, M. (2018). Is ecosystem health a useful metaphor? Towards a research agenda for ecosystem health research. In proceedings of the 17th IFIP Conference on

E-Business, E-Services and e-Society (I3E2018). Lecture Notes in Computer Science, Vol. 11159, Springer.
Hyrynsalmi, S., Suominen, A., & Mäntymäki, M. (2016). The influence of developer multi-homing on competition between software ecosystems. Journal of Systems and Software, 111, 119-127.
Iansiti, M., & Levien, R. (2004). Strategy as ecology. Harvard Business Review, 82(3), 68-81.
Islam, A.K.M.N., Mäntymäki, M. & Turunen, M. (2019) Understanding the Role of Actor Heterogeneity in Blockchain Splits: An Actor-Network Perspective to Bitcoin Forks. In Proceedings of the 52nd Hawaii International Conference on System Sciences. HICSS'52.
James, W. (1977). A pluralistic universe. Harvard, MA: Harvard University Press.
Kilduff, M., & Tsai, W. (2003). Social networks and organizations. Thousand Oaks, CA: Sage.
Kimble, H. J. (2008). The Quantum Internet. Nature, 453(7198), 1023.
Koskenvoima, A. & Mäntymäki, M. (2015). In proceedings of the 14th IFIP Conference on e-Business, e-Services and e-Society (I3E2015). Lecture Notes in Computer Science, Springer.
Leontjev, A.N. (1973). Activity and Consciousness. Revista Dialecta, 2(4):159–183
Lewin, R. (1999). Complexity: Life at the edge of chaos. Chicago, IL: University of Chicago Press.
Mian, S. Q., Mäntymäki, M., Riekki, J., & Oinas-Kukkonen, H. (2016). Social Sensor Web: Towards a Conceptual Framework. In the 15th IFIP Conference on e-Business, e-Services and e-Society (pp. 479-492). Lecture Notes in Computer Science, Springer, Cham.
Moore, J. F. (1993). Predators and prey: a new ecology of competition. Harvard business review, 71(3), 75-83.
Mäntymäki, M. (2008). Does e-government trust in e-commerce when investigating trust? A review of trust literature in e-commerce and e-government domains. In Towards Sustainable Society on Ubiquitous Networks (pp. 253-264). Springer, Boston, MA.
Mäntymäki, M., & Salmela, H. (2017). In Search for the Core of the Business Ecosystem Concept: A Conceptual comparison of business ecosystem. In 9th International Workshop on Software Ecosystems (IWSECO 2017) (p. 103).
Mäntymäki, M., Salmela, H. & Turunen M. (2018). Do Business Ecosystems Differ from Other Business Networks? The case of an emerging business ecosystem for digital real-estate and facility services. In proceedings of the 17th IFIP Conference on E-Business, E-Services and e-Society (I3E2018). Lecture Notes in Computer Science, Vol. 11159, (pp. 102-116). Springer:Cham.
Normann, R. (2001). Reframing business: When the map changes the landscape. John Wiley & Sons.
Normann, R., & Ramirez, R. (1993). From value chain to value constellation: Designing interactive strategy. Harvard Business Review, 71(4), 65-77.
Ocasio, W. (1997). Towards an attention‐ based view of the firm. Strategic Management Journal, 18(S1), 187-206.
Odum, E. P. (1966).The strategy of ecosystem development. Science, 164:81, 262-270.
Oh, D. S., Phillips, F., Park, S., & Lee, E. (2016). Innovation ecosystems: A critical examination. Technovation, 54, 1-6.
Pandey, A., & Gupta, R. K. (2008). A perspective of collective consciousness of business organizations. Journal of Business Ethics, 80(4), 889-898.
Peltoniemi, M., & Vuori, E. (2004). Business ecosystem as the new approach to complex adaptive business environments. In Proceedings of eBusiness research forum (Vol. 2, pp. 267-281).
Teece, D. J. (2007). Explicating dynamic capabilities: the nature and microfoundations of (sustainable) enterprise performance. Strategic Management Journal, 28(13), 1319-1350.

Teece, D. J. (2010). Business models, business strategy and innovation. Long Range Planning, 43(2-3), 172-194.
Tsoukas, H. (2017). Don't simplify, complexify: From disjunctive to conjunctive theorizing in organization and management studies. Journal of Management Studies, 54(2), 132-153.
Tsoukas, H., & Chia, R. (2002). On organizational becoming: Rethinking organizational change. Organization Science, 13(5), 567-582.
Turunen, M. (2014). Consciousness View of Organizations. In D.M. Boje and T.L. Henderson (Eds.) Being Quantum: Ontological Storytelling in the Age of Antenarrative, pp. 385-403 Newcastle Upon Tyne, UK: Cambridge Scholars Publishing.
Turunen, M. (2015). Toward a Consciousness-Based View of Organizing. Aalto University, pp. 1-220. Unigrafia: Helsinki.
Turunen, M. (2018). Storytelling on Consciousness-based View of Organizing. In D.M. Boje & M. Sanchez (Eds). The Emerald Handbook of Quantum Storytelling Consulting, pp. 73-83 Emerald: Los Angeles.
Vargo, S. L., & Lusch, R. F. (2010). From repeat patronage to value co-creation in service ecosystems: a transcending conceptualization of relationship. Journal of Business Market Management, 4(4), 169-179.
Vygotsky, L. S. (1980). Mind in society: The development of higher psychological processes. Massachusetts, MA: Harvard University Press.
Wirén, M., & Mäntymäki, M. (2018). Strategic Positioning in Big Data Utilization: Towards a Conceptual Framework. In proceedings of the 17th IFIP Conference on E-Business, E-Services and e-Society (I3E2018). Lecture Notes in Computer Science, Vol. 11159, Springer. In Conference on e-Business, e-Services and e-Society (pp. 117-128). Springer, Cham.
Xu, Y., Turunen, M., Ahokangas, P., Mäntymäki, M. & Heikkilä, J. (2018). Contextualized business model: The case of experiential environment and AI. 2nd Business Model Conference, Florence, Italy, June 6-7, 2018.

Engineering and Business Aspects of SaaS Model Adoption: Insights from a Mapping Study

Andrey Saltan[1,2] and Ahmed Seffah[1]

[1] Lappeenranta University of Technology, Lappeenranta, Finland
[2] National Research University Higher School of Economics, Moscow, Russia

andrei.saltan@lut.fi. ahmedryanseffah@gmail.com

Abstract. Transition to Software-as-a-Service (SaaS) model is revolutionizing the software market with significant impact on both SaaS providers and consumers. Research on business aspects of SaaS was usually limited to exploring adoption factors from a consumer perspective. Research on engineering aspects of SaaS focuses primarily on cloud computing architecture and software development. The prime aim of this paper is to assemble these engineering and business aspects of SaaS while building a better understanding of the servitization transformation of software companies as well as classifying, analyzing and putting in correspondence state-of-the-art studies focused on SaaS product and project management, business practices and engineering processes. The mapping study findings and interpretations are summarized to emphasize the significant research challenges regarding the required engineering and business efforts needed for the adoption service-oriented model. The performed study creates an appropriate basis for the further research.

Keywords: Software-as-a-Service, SaaS, Software Engineering, Software Business, Software Product Management, Systematic Mapping study

1 Introduction

Nowadays, Software-as-a-Service (SaaS) is a software licensing and delivery model frequently used by software companies both in B2B and B2C markets. The global public cloud services market is expected to grow 18% in 2017 to total $247 billion [42, 53]. These studies also predict Cloud Application Services or SaaS market segment to be ranked second with the capitalization close to $46 billion inferior just to Cloud Advertising.

SaaS as licensing and delivering model suppose that service consumers receive remote access to the software on a subscription basis rather than buying a software license and installing the software on their computers and servers. The software itself is owned, developed and managed by a service provider. The rational economic analysis of the SaaS model has persuaded developers as well as consumers to consider shifting toward the SaaS model. For software companies, the service-oriented model in comparison to traditional software product development offers benefits related to revenue and profit as well as customer acquisition and retention [59]. However, in both B2B and B2C markets, the new model also promises significant benefits for consumers ranging from cost reduction and scalability to remote access and compatibility [3]. These advantages ensured the rapid diffusion and adoption of the model on the market.

SaaS adoption has become the topic of interest for scientists in different research domains. It has gained considerable attraction in IT management as well as software business domains due to its capability to make a significant influence on business practices and software market structure [11]. The broad range of studies in this domain was mainly focused on the identification and evaluation core SaaS model adoption drivers and inhibitors, assessing economic benefits of the service-oriented model as well as propose solutions to further market growth challenges. At the same time, the SaaS paradigm, as well as overall cloud computing, has also been the center of attraction in software engineering and information technology research domains. Mostly often studies in these domains investigated engineering and technological challenges of cloud computing including among others issues of service-oriented architecture and cloud development methodologies [27].

Persistent efforts in these research domains, even being often performed without the necessary integration, made it possible to find solutions to the major challenges facing the development of emerging service-oriented model including consumers concern regarding SaaS adoption [3] and the transition of Software-as-a-Product (SaaP) companies to SaaS providers [60]. However, both technological and methodological transition in existing software companies took place without proper support and to what extent companies adopted the service paradigm remain unclear for the researchers. Even though SaaS market is proliferating driven by payoffs on both supply and demand sides, undiscovered aspects related to the adoption and acceptance of service-oriented model by SaaS providers can be real obstacles to an active and efficiently adoption of SaaS.

This paper presents the results of the systematic mapping study on adoption SaaS by software companies, mainly the factors affecting their readiness and acceptance of the SaaS model. This study investigates the role of technological/engineering and management/business aspects of SaaS adoption and proposes a taxonomy of factors that should be considered while re-engineering or establishing development process, product management practices and overall business model. The current study includes several contributions to software business and cloud computing research domains.

2 Background and Motivation

In the 2000s with the rapid development of the Internet, SaaS model as one of the cloud computers paradigm pillars started to supplant both the traditional SaaP model and the software outsourcing one called ASP [51]. The first attempts to explicitly define this model were able to convey the essential characteristics but were different regarding architecture [43]. Nowadays the most common and widely accepted definition is the one presented in 2011 by United States National Institute of Standards and Technology (NIST) as “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction” [44].

2.1 SaaS Model Adoption

A narrow interpretation of the SaaS as licensing and delivery model assumes considering it as process innovation. From an economic perspective, SaaS model is an example of innovation [13, 19] identified four categories of innovation by area of focus: product/service innovations, process innovations, organizational innovations, and technological innovations.

However, nowadays broader interpretation is more widely-accepted that recognizes the unique process, technological and organizational innovations behind this model [22]. SaaS model also comes with a radical shift of the means by which software is engineered and developed as well as its product strategy and pricing, lifecycle management, customer involvement, and relationship management. Besides different engineering and business practices and processes, the service-oriented paradigm requires new software architecture [57]. Given the above considerations, the service-oriented paradigm adoption process takes place across different dimensions that for the purpose of this study we will identify as engineering and business ones.

SaaS market is populated with both new companies, established initially as service companies, and companies that entered this market from the SaaP market with software services developed on the basis of existing software products [36]. Companies of the first type followed the “development from scratch for SaaS” approach considering features and capabilities of the SaaS model, while the ones of the second type were “re-engineering for SaaS” with the plan either to supplement the already existing on-demand software with specific SaaS solutions, or to implement the full transformation within some period [4]. Adaption of the SaaS model requires significant efforts and expertise from software companies in establishing or transforming their business modeling and re-engineering their business processes.

2.2 Previous studies and reviews

The literature on cloud computing adoption, acceptance and diffusion issues is growing fast. We were able to identify 34 reviews and mapping studies among the initially collected body of literature related to different issues of SaaS. [14, 23, 64, 67] observe and explore the overall research of cloud computing with SaaS as one of its pillars identifying characteristics and market structure as well as further development prospects and benefits for various market participants. Additionally, several reviews had a specific context like SMEs [49], healthcare [17] or education [15].

As SaaS model is much more demand-side driven compared to the traditional SaaP [61], a broad range of studies was devoted precisely to cloud computing and SaaS adoption issues. Reviews [3, 20, 46, 51] classify and assess legal, technical and managerial factors that have the most significant impact on this process.

We were not able to find any review that systematically discusses the adoption of SaaS model by software companies, although some aspects have been investigated sufficiently. [41] systematically review the existing software product management practices in companies developing cloud services. [4, 56, 57] reviews studies on SaaS development processes practices, while [34, 63] investigate software development methodologies widely used in SaaS development. Finally, [55, 66] focus on such business aspect of SaaS model adoption like value proposition and organizational integration respectively.

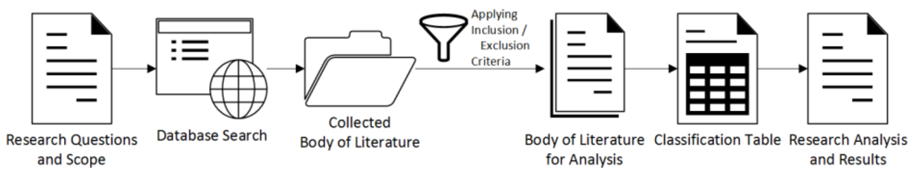
The mapping study goes beyond the existing studies and is intended to bring the same depth of analysis that exist in existing studies on the demand side to the issues of adoption SaaS model by service providers and developers. The review serves as a bridge between recently conducted studies regarding particular aspects of software business at the age of SaaS, and it provides a taxonomy of SaaS adoption challenges and factors facing service providers and considering existing interplay between the engineering and business aspects.

3 Research Methodology

Grounded in the protocol outlined from the guidelines [32, 47], the mapping study was conducted according to the process depicted in **Figure 1**. It focuses on aspects that characterize and define the adoption of SaaS by software companies across various research domains and types of studies. To get an integrated and comprehensive look into the area of SaaS adoption and distribution of existing research in it, this mapping study answers the following research questions:

- **RQ 1:** What are the critical factors regarding SaaS model adoption by software companies that have been identified in academic literature?
- **RQ 2:** What are the aspects of SaaS adoption and challenges facing software companies that have been addressed in the research studies?

Figure 1. Research process



The overall primary research scope is as follow:

- Population: We were looking for papers published since 2010. Publication venue was also limited to articles in peer-reviewed journals and conference proceedings.

- Intervention: Following approaches were implemented to obtain relevant mapping study: prime literature body was collected with further data extracted and analyzed in the form of answers to research questions.
- Comparison: The mapping study compares various issues and aspects regarding SaaS model adoption from multiple dimensions.
- Outcome: The primary body of literature covers a wide range of existing studies. Mapping study conducted upon the body of literature provides clear perceptions regarding SaaS adoption including critical challenges across multiple domains, significant trends in the further development of the concept both in industry and academy.

Given the large body of literature regarding the research topic, the data collection routine was based on automatic across multiple scientific databases and digital libraries. We defined primary search terms that could form the search queries for major scientific databases and libraries as well as set search limitations regarding time, publication type, and research area. The first list of terms included different possible synonyms for SaaS such as "software as a service", "cloud service", “web service”, “digital service” and “cloud computing”. The second one includes terms that can be used in papers related to different aspects of SaaS adoption by software companies: “software product management”, “business model”, “pricing”, “service architecture”, “service engineering”, “software development”, “deployment models”, “adoption” and “servitization”.

A manual one supplemented automatic search. Using looking back method (exploring reference list of already identified papers) as well as by looking forward one (analysis of papers which cited the already identified ones), we were able to collect additional studies relevant to the research scope. The Inclusion Criteria was applied while screening identified papers’ title, keyword and abstracts to evaluate the compliance with quality and availability criteria. Once the Inclusion Criteria was implemented, we explored full-text of the papers to determine the ones that provide a definite contribution to SaaS model adoption in software companies and exclude those that consider the adoption issues as a context or motivation for the research. The initial scope of literature was limited from 758 to 48 items by applying Inclusion/Exclusion Criteria.

Multistage formal content analysis process was implemented to extract a taxonomy of aspects, factors, and challenges that affect SaaS model adoption by software companies with further analyzing and reporting.

4 Results

The results of the research are presented in the form of answers to the research questions followed by a reflection on research findings.

4.1 Rationality of SaaS Model Adoption

The rationality factors regarding SaaS model adoption by software companies were identified in the answer of RQ1. The vast majority of found papers discuss the attractiveness of the SaaS model for software companies. They also most often highlighted concerns that are more challenges rather than real obstacles towards SaaS adoption.

Regardless of whether companies adopt SaaS by transiting from SaaP model, developing SaaS that complements existing solutions or creating a SaaS from scratch, most often the adoption process is based on Electronic Data Interchange (EDI) adoption model [9, 25]. This model suggests three significant groups of factors that influence the adoption decision and process: Benefits, Readiness, and Pressure. The identified factors fully correspond to the logic of the EDI model. They are presented in **Table 1.**

Table 1. Factors of SaaS model adoption

Factor	Source
Business and Engineering Benefits	
Obtain long-term higher profit margin and revenue	[38, 40, 65]
Reduce opportunity costs and utilize the economy of scale benefits	[18, 33, 38]
Expand the range of services provided	[16, 38]
Obtain market leadership and changing “rules of the game”	[30, 31, 38]
Increase the reliability of customers relationships	[40, 54, 65]
Facilitate the geographical expansion	[18, 54]
Facilitate upgrading, modification and customization processes	[57, 65]
Obtain agility and scalability of the development and deployment	[27, 45]
Obtain better quality of business analytics for decision-making	[21, 48]
Avoid to a certain extent the software piracy	[38, 50]
Organizational and Technological Readiness	
Financial resources availability and investors’ readiness for changes in revenue structure	[54]
Investors, top-managers, and personnel intention	[30]
Availability of personnel with required competencies	[30, 31]
Internal and External Pressure	
Competitors pressure	[31]

4.2 Aspects and Challenges of SaaS Adoption

Publications regarding aspects of SaaS adoption and challenges facing software companies (**RQ2**) shows dissimilar results as most of the papers do not explicitly property formulate them. The findings are grouped into three subcategories, namely: Business Model, Operations, Product and Project Management as well as Technology and Engineering Practices. **Error! Reference source not found.** summarizes the adoption aspects and the underlying challenges.

Table 2. Aspects and Challenges of SaaS adoption

Aspect	Challenges	Source
Business Model and Strategic Product Management		
Business model and strategy	Redesign the business model and strategy to address such issues as the economy of scale, servitisation, and shift from on-premises to on-demand	[29, 31, 33, 35, 58]
Value proposition	Ensure that pricing scheme provides a sufficient level of flexibility and total cost reduction for consumer	[2, 6, 33, 66]
Value proposition	Provide customers with tangible arguments about the SaaS benefits including cost reduction, performance increase and share best practices	[6, 24, 36, 54, 66, 68]
Cost and revenue structure	Design flexible and value-based pricing policy based on subscription and pay-per-use models	[5, 8, 12, 16]
Customer segmentation and relationship management	Redefine the customer segments to address geographical expansion and deeper customization	[54]
Customer segmentation and relationship management	Provide customers with strong arguments against prejudices about high risks of security and privacy while using SaaS	[6, 39]
Distribution channels	Focus on using the direct Internet and platforms-related channels as major communication, distribution and sales channels	[16, 54]
Partners collaborations	Reconsider partners' network to address the new structure of distribution channels and include PaaS and IaaS providers	[12, 18, 54]
Legal affairs	Design reliable SLA and ensure its compliance with the SaaS model	[6, 12, 26, 66]
Legal affairs	Ensure that data storage and processing following the legislative regulations that may vary across countries and industries	[26, 66]
Tactical Product and Project Management		
Transformation and integration management	Design the process through which the SaaS adoption will take place	[10, 30, 31, 65]
Transformation and integration management	Design incentives policies and organize personnel training to overcome personnel resistance	[65]
Project schedule management and monitoring	Reconsider metrics to plan SaaS development projects and track performance and implementation of the SaaS model	[34]
Project and product resources management	Integrate SaaS infrastructure and related resources in the existing resource management practices	[54]

Customer support and services	Develop a higher level of integration between consumer and provider to the required level of quality	[37, 54, 66]
Customer support and services	Ensure proper data-recovery, data-migration, billing and auditing processes	[52, 66]
Operations, Technology and Engineering Practices		
Architecture and infrastructure	Ensure technical level of data security to mitigate risks of data loss and manipulation	[1, 6, 39, 66, 68]
Architecture and infrastructure	Ensure technical level of data privacy to mitigate risks of unauthorised access	[1, 6, 26, 66, 68]
Architecture and infrastructure	Adopt service-oriented and multi-tenancy architectures (technology, principles)	[12, 27, 28, 54, 61, 62]
Architecture and infrastructure	Ensure availability of the SaaS as this is one of the most critical issue for the consumers	[7, 26, 66]
Architecture and infrastructure	Ensure high level of interoperability that allow integration with other IT-systems and services	[7, 26, 66, 68]
Requirements management and quality assurance	Adopt new principles to involve customers in requirements prioritisation	[57]
Requirements management and quality assurance	Finding a balance between providing standard solutions and satisfying customized demand	[26, 38, 66]
Development practices and processes	Adopt agile software development methodologies with corresponding metrics	[4, 34, 54]
Testing and Maintenance	Update development and management processes including testing, release, and maintenance that are based on the principles of customers' involvement and collaboration	[31, 54, 57]

5 Discussion and Conclusion

The prime objective of the ongoing research presented in this paper is to address the interplay between business and engineering aspects of the SaaS model adoption. Companies were able to extend or establish a new business SaaS model and product management. However as highlighted in this paper, several adoption challenges may compromise the promises of the SaaS. The comprehensive systematic analysis of the literature aims to gain a deeper understanding of the factors, aspects, and challenges of SaaS adoption both the business and engineering ones. While answering the two research questions, the study makes two principal contributions and draw a roadmap for a research agenda on how to address the SaaS model adoption.

Regarding **RQ1**, it appeared that SaaS adoption is driven mostly by competitive pressure and expectations of potential benefits rather than consumers demand and partners pressure. Moreover, achieving the expected benefits requires significant efforts in consumers' and partners' relationship management, many of which are highly doubtful regarding SaaS. A little is known about the various aspects of organizational readiness for SaaS adoption.

RQ2 reveals a wide range of challenges discussed in the academic literature. From the business side the most frequently discussed challenges are related to the business model design, value proposition, and customer relationship management. From the engineering perspective, fundamental challenges were identified including the required security, privacy, scalability, and availability of the SaaS. The study also revealed that the literature appeared to be scarce and scattered regarding organizational as well as product and project management aspects of SaaS adoption. This situation is a real obstacle to achieving the advantages of the new model.

The research conducted may have certain limitations. First, providing a taxonomy of factors and challenges is not enough. We also need to study the interdependencies, mediation and moderation effects. It also needs to prioritize the identified factors and challenges. Second, the research should be extended to include a large number of studies from various domains and companies.

Acknowledgments

The first author has been partially supported by the Academy of Finland Grant no 310827 (MobiSPSULUT)

References

1. Ali, M. et al.: Security in cloud computing: Opportunities and challenges. Information Sciences. 305, 357–383 (2015).
2. Alshamaila, Y. et al.: Cloud computing adoption by SMEs in the north east of England: A multi-perspective framework. Journal of Enterprise Information Management. 26, 3, (2013).
3. Asatiani, A.: Why cloud? - A review of cloud adoption determinants in organizations. In: ECIS Proceedings. pp. 1–17 (2015).
4. Baliyan, N., Kumar, S.: Towards software engineering paradigm for software as a service. In: International Conference on Contemporary Computing (IC3) Proceedings. pp. 329–333 (2014).
5. Baur, A.W. et al.: Customer is King? A Framework to Shift from Cost- to Value-Based Pricing in Software as a Service: The Case of Business Intelligence Software. In: Conference on e-Business, e-Services and e-Society Proceedings. pp. 1–13 (2014).
6. Benlian, A., Hess, T.: Opportunities and risks of software-as-a-service: Findings from a survey of IT executives. Decision Support Systems. 52, 1, 232–246 (2011).
7. Bokhari, M.U. et al.: Cloud Computing Service Models: A Comparative Study. IEEE International Conference on Computing for Sustainable Global Development, INDIACom. 16–18 (2016).
8. Chun, S.-H., Choi, B.-S.: Service models and pricing schemes for cloud computing. Cluster Computing. 17, 2, (2014).
9. Chwelos, P. et al.: Research Report: Empirical Test of an EDI Adoption Model. Information Systems Research. 12, 3, 304–321 (2001).
10. Clegg, B. et al.: Transformation of a small-to-medium-sized enterprise to a multi-

organisation product-service solution provider. International Journal of Production Economics. 192, 81–91 (2015).
11. Cusumano, M.A.: The changing software business: Moving from products to services. Computer. 41, 1, 20–27 (2008).
12. D'souza, A. et al.: Software-as-a-Service: Implications for Business and Technology in Product Software Companies. In: PACIS Proceedings. (2012).
13. Damanpour, F., Gopalakrishnan, S.: The dynamics of the adoption of product and process innovations in organizations. Journal of Management Studies. 38, 1, 45–65 (2001).
14. Durao, F. et al.: A systematic review on cloud computing. The Journal of Supercomputing. 68, 3, 1321–1346 (2014).
15. El-Gazzar, R.F.: A Literature Review on Cloud Computing Adoption Issues in Enterprises. In: International Working Conference on Transfer and Diffusion of IT Proceedings. pp. 214–242 (2014).
16. Erek, K. et al.: Common Patterns of Cloud Business Models. In: AMCIS Proceedings. (2013).
17. Ermakova, T., Fabian, B.: Secret sharing for health data in multi-provider clouds. In: IEEE International Conference on Business Informatics, CBI. pp. 93–100 IEEE (2013).
18. Ferri, L. et al.: Analyzing Cloud-based Startups: Evidence from a Case Study in Italy. International Business Research. 10, 5, 73 (2017).
19. Gopalakrishnan, S., Damanpour, F.: Patterns of generation and adoption of innovation in organizations: Contingency models of innovation attributes. Journal of Engineering and Technology Management. 11, 2, 95–116 (1994).
20. Haag, S., Eckhardt, A.: Organizational cloud service adoption: a scientometric and content-based literature analysis. Journal of Business Economics. 84, 3, 407–440 (2014).
21. Hartmann, P.M. et al.: Capturing Value from Big Data - A Taxonomy of Data-driven Business Models Used by Start-up Firms. International Journal of Operations & Production Management. 36, 10, 1382–1406 (2016).
22. Heart, T. et al.: Software-as-a-Service Vendors: Are They Ready to Successfully Deliver? In: International Workshop on Global Sourcing of Information Technology and Business Processes. pp. 151–184 (2010).
23. Hoberg, P. et al.: The business perspective on cloud computing - A literature review of research on cloud computing. In: AMCIS Proceedings. 5 (2012).
24. Hsu, P.-F. et al.: Examining cloud computing adoption intention, pricing mechanism, and deployment model. International Journal of Information Management. 34, 4, 474–488 (2014).
25. Iacovou, C.L. et al.: Electronic Data Interchange and Small Organizations: Adoption and Impact of Technology. MIS Quarterly. 19, 4, 465–485 (1995).
26. Janssen, M., Joha, A.: Challenges for adopting cloud-based software as a service (SAAS) in the public sector. In: ECIS Proceedings. 80 (2011).
27. Joachim, N.: A Literature Review of Research on Service- Oriented Architectures (SOA): Characteristics, Adoption Determinants, Governance Mechanisms, and Business Impact. In: AMCIS Proceedings. 339 (2011).
28. Kabbedijk, J. et al.: Multi-tenant Architecture Comparison. In: European Conference on

Software Architecture. pp. 202–209 Springer, Cham (2014).
29. Kaltencker, N. et al.: The disruptive potential of software as a service: Validation and application of an ex-ante methodology. ICIS Proceedings. 3808–3826 (2013).
30. Kaltenecker, N.: Managing Disruptive Change: Successful Transformation from On-premises to SaaS in B2C Software Companies. PACIS Proceedings. (2015).
31. Kaltenecker, N., Hess, T.: From on-Premises To on-Demand: Learning From Two Cases of Transformation of Software Companies. ECIS Proceedings. 1–15 (2014).
32. Kitchenham, B.A. et al.: Using mapping studies as the basis for further research - A participant-observer case study. Information and Software Technology. 53, 6, 638–651 (2011).
33. Kranz, J.J. et al.: Understanding the influence of absorptive capacity and ambidexterity on the process of business model change – the case of on-premise and cloud-computing software. Information Systems Journal. 26, 5, 477–517 (2016).
34. Kupiainen, E. et al.: Using Metrics in Agile and Lean Software Development - A Systematic Literature Review of Industrial Studies. Information and Software Technology. 62, 1, 143–163 (2015).
35. Labes, S. et al.: Success Factors of Cloud Business Models. In: ECIS Proceedings. pp. 1–14 (2015).
36. Labes, S. et al.: Successful Business Model Types of Cloud Providers. Business and Information Systems Engineering. 59, 4, 223–233 (2017).
37. Lewandowski, J. et al.: SaaS Enterprise Resource Planning Systems: Challenges of Their Adoption in SMEs. In: IEEE International Conference on e-Business Engineering. pp. 56–61 (2013).
38. Liao, H.: SaaS Business Model for Software Enterprise. In: IEEE International Conference on Information Management and Engineering (ICIME) Proceedings. pp. 604–607 (2010).
39. Loske, A. et al.: Perceived it security risks in cloud adoption: The role of perceptual incongruence between users and providers. In: ECIS Proceedings. (2014).
40. Luoma, E.: Examining Business Models of Software-as-a-Service Firms. In: International Conference on Grid Economics and Business Models. pp. 1–15 Springer, Cham (2013).
41. Maglyas, A. et al.: What do we know about software product management? - A systematic mapping study. In: International Workshop on Software Product Management, IWSPM. pp. 26–35 (2011).
42. Mahowald, R.P. et al.: Worldwide Software as a Service and Cloud Software Forecast, 2017-2021. (2017).
43. Mäkilä, T. et al.: How to define Software-as-a-Service - An empirical study of Finnish SaaS providers. In: International Conference of Software Business Proceedings. pp. 115–124 (2010).
44. Mell, P.M., Grance, T.: The NIST Definition of Cloud Computing. (2011).
45. Misra, S.C., Mondal, A.: Identification of a company's suitability for the adoption of cloud computing and modelling its corresponding Return on Investment. Mathematical and Computer Modelling. 53, 3–4, 504–521 (2011).
46. De Paula, A.C.M., De Figueiredo Carneiro, G.: Cloud computing adoption, cost-benefit relationship and strategies for selecting providers: A systematic review. In: Proceedings

of the International Conference on Evaluation of Novel Software Approaches to Software Engineering. pp. 27–39 (2016).
47. Petersen, K. et al.: Systematic Mapping Studies in Software Engineering. In: Proceedings of the International Conference on Evaluation and Assessment in Software Engineering. (2008).
48. Resceanu, I.C. et al.: SaaS solutions for small-medium businesses: Developer's perspective on creating new SaaS products. In: International Conference on System Theory, Control and Computing (ICSTCC) Proceedings. pp. 140–144 (2014).
49. Rodrigues, J. et al.: Software as a Service Value and Firm Performance - A literature Review Synthesis in Small and Medium Enterprises. Procedia Technology. 16, 206–211 (2014).
50. Saltan, A. et al.: A dynamic pricing model for software products incorporating human experiences. In: International Conference on Software Business (ICSOB) Proceedings. pp. 135–144 (2016).
51. Schneider, S., Sunyaev, A.: Determinant factors of cloud-sourcing decisions: Reflecting on the IT outsourcing literature in the era of cloud computing. Journal of Information Technology. 31, 1, 1–31 (2016).
52. Da Silva, E.A.N., Lucrédio, D.: Software Engineering for the Cloud: A Research Roadmap. In: Brazilian Symposium on Software Engineering (SBES) Proceedings. pp. 71–80 (2012).
53. Smith, D.M. et al.: Predicts 2017: Cloud Computing Enters Its Second Decade. (2016).
54. Stuckenberg, S. et al.: The impact of software-as-a-service on business models of leading software vendors: experiences from three exploratory case studies. In: PACIS Proceedings. Paper 184 (2011).
55. Stuckenberg, S. et al.: Understanding the role of organizational integration in developing and operating Software-as-a-Service. Journal of Business Economics. 84, 8, 1019–1050 (2014).
56. Stuckenberg, S., Heinzl, A.: The impact of the software-as-a-service concept on the underlying software and service development processes. In: PACIS Proceedings. Paper 125 (2010).
57. Stuckenberg, S., Stefan, B.: Software-as-a-service development: Driving forces of process change. In: PACIS Proceedings. Paper 122 (2012).
58. Su, N.: Emergence of Cloud Computing: An Institutional Innovation Perspective. In: ICIS Proceedings. pp. 1–12 (2011).
59. Suarez, F.F. et al.: Services and the Business Models of Product Firms: An Empirical Analysis of the Software Industry. Management Science. 59, 2, 420–435 (2013).
60. Suresh, S., Ravichandran, T.: Firm transitions from products to services and mode of entry. In: HICSS Proceedings. pp. 1400–1409 (2015).
61. Tariq, A. et al.: Requirements Engineering process for Software-as-a-Service (SaaS) cloud environment. In: Proceedings of the International Conference on Emerging Technologies (ICET). pp. 13–18 (2014).
62. Tsai, W.T. et al.: Software-as-a-service (SaaS): Perspectives and challenges. Science China Information Sciences. 57, 5, 1–15 (2014).
63. Tuli, A. et al.: Empirical Investigation of Agile Software Development: A Cloud Perspective. ACM SIGSOFT Software Engineering Notes. 39, 4, 1–6 (2014).

64. Venters, W., Whitley, E.: A Critical Review of Cloud Computing: Researching Desires and Realities. Journal of Information Technology. 27, 3, 179–197 (2012).
65. Vial, G., Rivard, S.: From On-premises Software to Software-as-a-Service: Transforming C3 Solutions. International Journal of Case Studies in Management. 15, 3, 1–18 (2017).
66. Walther, S. et al.: Success factors and value propositions of software as a service providers - A literature review and classification. In: AMCIS Proceedings. 1 (2012).
67. Yang, H.-I. et al.: A Framework for Service Morphing and Heterogeneous Service Discovery in Smart Environments. In: International Conference on Smart Homes and Health Telematics Proceedings. pp. 9–17 (2012).
68. Zhang, Y. et al.: Converting legacy desktop applications into on-demand personalized software. IEEE Transactions on Services Computing. 3, 4, 306–321 (2010).

The Open Source Software Business Model Blueprint: A Comparative Analysis of 10 Open Source Companies

Zeena Spijkerman and Slinger Jansen

Utrecht University, the Netherlands
Corresponding Author: slinger.jansen@uu.nl

Abstract. The success of open source software business models is currently not well understood, leading to poor investment decisions and forcing entrepreneurs to make the same mistake repeatedly. In this paper, we report the results of 10 comparative case studies of open source businesses, using the Software Business Model Framework as the underlying conceptual model. The extracted findings are summarized in a referential business model blueprint and a series of lessons for entrepreneurs and investors. With these lessons entrepreneurs can prevent commonly made mistakes and investors can profile potentially successful companies.

1 Introduction

Starting from the mid-90s, several small open source businesses changed the software industry by offering a cooperatively produced collective good instead of proprietary software [1]. Since then, there has been increasing interest among academics and practitioners in Open Source Software (OSS) [2]. Initially created by the hacker movement [3], the OSS phenomenon has now metamorphosed into a more mainstream and commercially viable product [4] with ground rules defined by the Open Source Initiative [5]. When companies recognized this new type of software as a way to generate revenue, new business models arose.

OSS was starting to be used as a new business strategy to reduce costs and make maximum profits by a large stream of software companies [6]. These companies became well-known by offering cooperatively produced software [1] for free within an already existing corporate market. Due to this cooperative approach to product development, open source is often not seen as a business approach but more as a technology model [6, 7]. Nevertheless, different types of business models have been applied where different types of stakeholders and external factors are playing a role in the start-up phase of OSS producers, distributors and service providers. It is however unclear how these business models are formulated and whether they are unique or contain patterns. Therefore, the research question of this work is “How can a business model blueprint be created for future open source software businesses?”.

Currently, companies are focused on the Open Source concept and create considerable revenues through open source software and services [6] but with different offering profiles. An example is Red Hat ®and Linux who not directly make money from the open source programs but price complementary services [8]. The OSS business model and its revenue logic are not always as obvious to perceive [9] and there is little to no research performed that looked into the building blocks of successful OSS business models. A large amount of business literature is devoted to the definition of a business model, but this research specifically focuses on OSS businesses and will therefore use a more open source oriented definition of a business model and its characteristics.

Onetti et al. [10, p.224] recognize the lack of a comprehensive theoretical framework about OSS business models and that this is due to the relative newness of the phenomena. This paper contributes to the field by exploring the way open source businesses have entered the software ecosystem and created a foundation for following companies. Additionally, knowledge is contributed to the software business domain by looking specifically at the current business models of B2B focused OSS businesses. A comparative case study is conducted by reviewing the business models and factors for success of 10 OSS businesses. The business models are defined and conceptualized by application of the Software Business Model Framework of [11]. The outcomes of a literature study around the subjects of OSS consortia, their business models, the value exchanges within them, and how success could be measured, are used for a thorough analysis of the determinants. These interviews with expert employees provide information about the success of these OSS businesses and will be used to create an OSS business model blueprint and a guideline for OSS start-ups.

The body of knowledge in OSS research lacks focus on the building blocks constructing business models of successful OSS businesses. This research tries to establish the determinants that make OSS businesses thrive by looking at previous success-stories. In this research the definition of OSS businesses as described by [12] will be used: firms that supply, in various ways, open source based products and services and release them under Open Source licenses. This definition will simultaneously be used with the Open Source Definition, originally acquired from the Debian free Software Guidelines. We aim to further investigate and build a foundation for both OSS and business literature. Literature based on empirical data focusing on OSS entrepreneurs is scarce. This leaves this vulnerable group of starters without sufficient guidelines while entering an upcoming market.

The paper provides the following contributions:

– In Section 3 an **OSS Business Blueprint** is provided, using the e3-value modelling language, that shows the main participants in an OSS business network.
– Section 4 provides insight into 10 case studies of open source businesses and how they are currently implementing the Software Business Model Framework of Schief.
– Section 6 provides advice for OSS entrepreneurs and startups, which can be summarized into finding an appropriate market with a differentiated product, using existing libraries and open source project for your proposition, and find

ways to extract value from the market, typically by offering dual licensing software and support contracts.
– Finally, in Section 6 we extract an OSSS business model blueprint for open source companies and startups.

2 Research Approach

For this qualitative research, a multiple-case study is selected because it enables the exploration of phenomena within, in this case, the open source ecosystem [13]. The use of a variety of data sources ensures that the unknown field is explored to eventually determine the explicit success determinants of OSS business models. The primary data used for this research is collected from a comparative case study which and is backed by a literature study. Subsequently, internal validation is performed by the interviewees.

2.1 Case Study

The source of evidence is based on individual depth interviews [14] within the sample of OSS companies. The final determinants of success rest on a comparative case analysis of the interview transcripts of 10 OSS companies which are chosen based on pre-determined sample criteria. The case study approach is based on the three phases of the Case Study Protocol(CSP) constituted by [15] which is based on research by Eisenhardt et al. [16]. The authors describe the CSP as a guideline for data analysis containing the procedures for conducting research an is also used as a research instrument.

The second stage of phase one of the CSP depicts the selection of the cases where a specified population is defined. Therefore in this section we have predefined sample criteria for the selection of the cases. For case studies to give significant results, random selection of the sample is neither necessary, nor preferable [16]. The size of the sample for this research is controlled by theoretical and practical considerations [17]. Theoretically, the size of the sample influences the generalizability of the research, therefore a big sample size (around 20) is preferable. Practically, by convenience sampling the sample size is smaller. This is due to response time of the open source companies and the number of interviews that have to be performed within the time constraint of this research.

The sample consists of OSS businesses that are chosen based on particular search criteria. We are interested in companies that follow a certain quality standard in their business and share the interest in OSS. The OSS company that fits within the sample;

1. is registered as a company,
2. is B2B; meaning the business is providing OSS to other businesses,
3. is a software vendor; meaning it creates and offers (open source) software,
4. made the code of the software freely available,
5. hosts an open source community,
6. generates revenue.

In order to develop software under the Open Source name there are some requirements as mentioned in the Open Source Definition as stated in [5]. This means that besides the company criteria chosen by the authors, the companies should follow the rules set by The Open Source Definition.

Phase two of the CSP characterizes the iterative data collection and analysis, which in this research starts by conducting interviews within the sample. The expert-interviews are a combination of open questions and the predefined elements from the Software Business Model Framework of [18]. The interviews are semi-structured and held with practitioners in the OSS field. The participants of the interviews are chosen because of their experiences which reflects the scope [19] and their ability to answer the interview questions. The interview comprises two parts. The first part is based on 10 open questions focused on the background of the company, the entrepreneurial aspects, and the interviewees' view on success. The second part of the interview is based on the work of [18] and gives insight in the particular characteristics of the analyzed business models.

The second stage of phase two of the CSP is the analysis of data within- and cross-case. The interviews are recorded and transcribed to eventually be analyzed with the NVivo tool (see [20] for more information). The answers of the stakeholders are coded within the tool based on the categorized questions and SBMF components.

This comparative case study is based on the comparison of the completed SBMFs by assembling all of them in a single table. Altogether the data is analyzed to derive a blueprint for Open Source start-ups. The data is analyzed to perform the third step of the CSP where the findings are used to sharpen the construct definitions. Moreover, in this stage the data is internally validated by the interviewees. Due to specific business model information the outcomes of this research are anonymised. Table 1 shows the profiles of the interviewees of the comparative case study in random order.

The external validity of this work can be challenged, as a relatively small number of case studies was included and the research is based on a convenience sample. The companies are successful and some of them have gone through an Initial Public Offering. However, the small number of cases does not give any guarantee that the blueprint is a formula for success in open source business. That in effect is also not the goal: we mainly aim to present the current status of open source business models in the field.

Case	Type of software	Founded	HQ	FTE	Interviewee role
A.	Project Management Tools	2015	Spain	10-50	CEO
B.	Integration platform and ESB	2006	U.S.A.	1,000-2,000	Dev Manager
C.	Linux distribution	1992	Germany	1,000-2,000	Regional Director
D.	Application Service Provider	2003	NL	10-50	Founder
E.	Git-repository management	2011	No main office	250-1,000	Product Manager
F.	Government geographic data publishing	2007	NL	10-50	Software Engineer
G.	Content Management System	2016	Germany	10-50	CEO
H.	ERP+CRM	2001	U.S.A.	50-250	CEO
I.	Domain Name system server	1999	NL	250-500	Product Manager
J.	Linux distribution	1993	U.S.	10,000-15,000	Accout manager

Table 1. Case study company details

3 Conceptual Models

A method used to define the characteristics of software business models is the Software Business Model Framework (SBMF) [11]. They state that a business model is composed of a number of strategy elements, and that their model make the strategic choices explicit. The Software Business Model Framework is composed of 5 groups that in turn contain 20 elements that are recommended as guidelines to characterize a business model [18]. The 5 groups are based on an extensive literature research and come together as: strategy, revenue, upstream, downstream and usage. The framework is used to analyze and perform the comparative case analysis. The use of this framework enables us to compare the business models of OSS companies on the same level and define the determinants of success.

We also define a value model to create understanding of what characterizes OSS business. First, the *actors* or *market segments* exchanging value in a business model are defined. Following the guidelines of the e3 value model of [21, p.48], "an actor is perceived by his/her environment as an economically independent (and often also legal) entity". The authors define the market segment as a: "concept that breaks a market (consisting of actors) into segments that share common properties" . The following actors and market segments are recognized who each execute activities:

– **Developers**: The developers write the code as the base of an OSS product. Additionally, they offer free support through the OSS community.
– **Investors**: Investors play a fundamental role for OSS start-ups and non-profit foundations offering funding for the development of OSS.

– **Customers**: The customers are the end-users of the OSS and purchase the product and close services and/or support contracts with OSS vendors.
– **OSS vendor**: The OSS vendor providing the product, services and/or support.
– **Foundation/ Association**: The non-profit software foundation works as a collaboration enabler between the OSS community and the commercial OSS vendor [22].
– **OSS community**: The community operates like a hub since it directs the value directed towards the OSS vendor, the customers and possibly an OSS foundation.

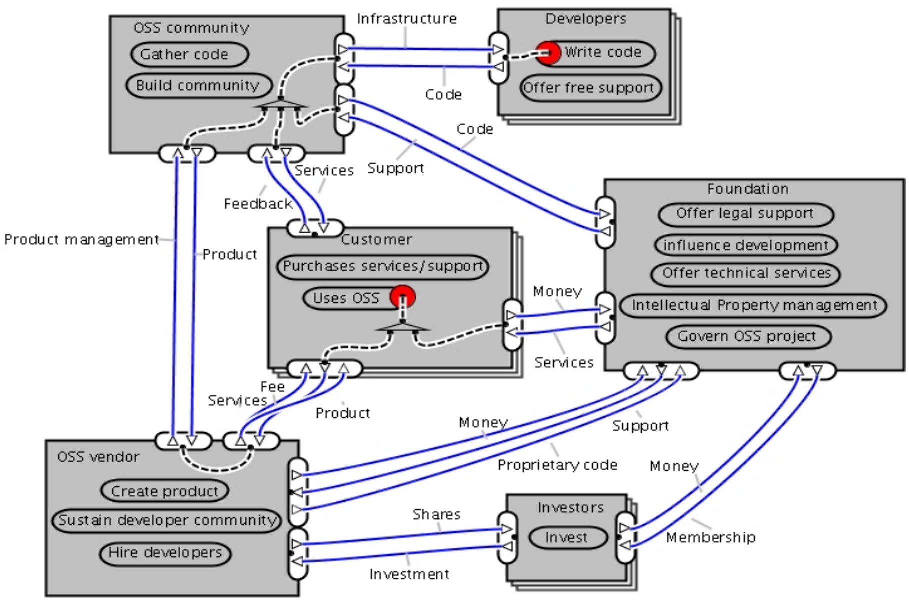


Fig. 1. E3-value model of OSS business model stakeholders and their value exchanges.

Figure 1 shows a value exchange between the community and the OSS vendor where the community exchanges the value object "product" since they develop a product based on the value object "code" offered by developers. According to [23], the community creates the value by developing a product for the OSS vendor leveraging a faster time-to-market and low development costs. In return, the OSS vendor helps the community to market it and offers "product management" to make it marketable as a professionally developed product [23]. Additionally, the OSS vendor often provides a platform for the OSS community to communicate and exchange code in order to sustain the community activities. OSS vendors might receive monetary investments from investors in exchange for company shares. These vendors take advantage from the tight connection with the OSS community by hiring the best developers from this community to work full-time on their product.

The ”foundation” actor offers developers several services like legal support and intellectual property management. The foundation is a separate entity which is able to host and govern a software project [24] when the vendor is offering the services to the end customer. The company exchanges value in return in the form of sponsorships and support to both the community and/or a foundation [22]. [22, p.408] explains that assets like proprietary code, financial resources and hardware can be donated by the vendor to a foundation, and in return some foundation offer a membership as a sponsor with an advisory role. The same role is apparent for investors who want to support OSS foundations by offering money or intellectual property. The customer has two possibilities when it comes to choosing the source of support and/or services. The OSS vendor offers paid support and/or services while the OSS community offer this for free. It is up to the customer whether they want to sign a SLA with a vendor or to find help from the community. Here, the foundation often works as a middle-man between the customer and the community. The OSS community receives value from the feedback given by the customers which in turn makes it possible to offer the users and vendor a better product.

4 Case Study Analysis

The data compelled from the interviews is based on the five building blocks of Schief’s [11] Software Business Model Framework: Strategy, Revenue, Upstream, Downstream and Usage.

Strategy The value proposition is aiming at the competitive advantage of a company’s offering. Three companies agree on the importance of being an innovation leader, which is achieved by offering new and disruptive software and services [11, p.72]. Three companies focus on the functionality of their product with its available features. The value proposition on which most companies agree on is the quality of their product, meaning that they aim for high consistency and dependability. Being an innovation leader in the open source ecosystem means that even under conditions of market failure, innovation is achieved by the collaboration of the community working on a public good [1]. Hippel and Krogh see that OSS business models ”present a novel and successful alternative to conventional innovation models” (p.212). Only one company has a value proposition focused on their image in the market or tries to differentiate through their price-scheme. In the *investment horizon* there is a clear time strategy that most companies use: the growth model. According to [25, p.731] the growth model is based on a start-capital including investment with later on a reinvestment in order to grow the value of the firm. This model aims at growing to achieve capital gain for new investors.

Interviewee of company F. addresses that they want to grow in number of employees but do not have a clear strategy to do so. Company E. on the other hand, has a clear strategy to grow in terms of revenue, trying to go public at the end of 2020. Company B. also measures growth in terms of revenue, wanting to have generated a billion dollar revenue for 2020. Company I. also tries to grow but just to meet the vision of the company and not to create such revenues. Company D. envisions growth in the number of partners and users. Usage is also measured by company G. which is trying to acquire more users to be visible in the market and to advertise their services. Company F. uses the Income model, working up to a point where they can sustain the business without making a lot of profit. Only company G. follows a social model, meaning that they are not focused on making profit but on specific clients such as governmental- and other non-commercial institutions.

Revenue For all companies the revenue source is directly paying customers. This is partially due to the fact that service agreements are offered for long-term and based on direct sales [26]. %50 of the companies use a hybrid combination of both usage-based and usage-independent pricing. Company B. bases their prices on the through-put when the platform is used. Company I. bases the price on the number of subscribers which the client connects to their product. The majority of the companies have chosen for recurring payments, similar to subscription fees. Explained by the interviewee of company I., a subscription model is necessary because developers have to be paid on a steady base. Besides that, companies need a constant cash-flow to pay the partners and for other services. Only one company has an upfront payment flow structure, and two have a hybrid combination of upfront and recurring structures. Company E. states that it is very motivating for the sales department to close large deals, and will even try to offer multi-annual contracts.

Upstream The *Upstream* component defines the development of a product and/or service. Many of the companies provide cloud computing solutions which gives a clear view on the current trends in the software ecosystem. As this research is focused on open source business models it is expected the companies apply either Copyleft Licenses (e.g. GPL), Permissive Licenses (e.g. BSD) or a dual model of both [11]. Six out of ten companies use the GPL license for their software, this corresponds with the fact that GPL is the most used license for OSS [27].

Four of the OSS companies offer both open source solutions and a proprietary solution next to each other. This complies with the dual license business model where a company markets the software product with the choice of either an open source licensed software product or a commercial/proprietary one [28] as described in section. Five companies produce their software in bulk and four in batch. Company E. addresses that they never make something for one customer. When the customer indicates that they want a certain function, the interviewee asks which problem they want to see fixed and sees if they can create something that is interesting for more clients, so in bulk.

Eight out of ten companies mostly spend money on Research & Development, so the personnel costs for the development of the software. The Interviewee of company G. states that they hire developers from the open source community to spend a certain amount of their time on the product. The interviewee says that the developers like this because they usually use their spare time to write code but now use that time to spend it with their families. This shows that OSS companies use free labour of the community and combine this with the work of in-house developers. They do this to ensure that the project is not dependent on the work of the community. Besides the research and development costs, companies B., G. and J. state that they also spend a lot of money on marketing.

Downstream The *Downstream* component is focused on the customers by characterizing the target market and distribution areas. The target customer for the companies are typically large organizations. Company E. explains that as a B2B company you earn the most money with offering your product to large clients. That is also why many of the OSS products have an open source solution and a separate enterprise product to offer companies that are willing to pay. Interviewee A. states that you need success with smaller companies to convince the larger ones, therefore the company is currently focusing on small and medium sized companies. The *target industry* component is obtained from the Standard Industrial Classification [29]. The option "All", which is mostly picked, conceptualizes companies which sell horizontal solutions that can be applied in any chosen industry [11]. Next to the options provided by Schief, most companies admit being very active in the field of digital marketing through online advertisement.

Usage The final component *Usage* concerns how the software can be offered, implemented, and used. The *implementation effort* is based on the effort that is required to install and configure the software. The software of the stakeholders seems mostly to be medium since the customers will not always be able to install the software themselves and might need some assistance. Some of the interviewees admit that especially the open source versions seem to be harder to install if the user has no prior knowledge of software. The *operating model* on which the software is deployed differentiates between two main deployment models: on premise and on demand. The hybrid combination of both on premise and local systems is the most often chosen option.

A company's *support model* depends on the support contracts signed by the customer [30]. Standard support is the same for all customers while the customer specific support model offers a customized individual support contract. The majority of the sample offers different types of support contracts. This mainly has to do with the fact that OSS companies cannot ask money for the software, therefore business models of OSS companies are primarily built around software related services [31].

5 Suggestions for OSS Entrepreneurs

It is possible to create an OSS start-up without having an existing open source community. We have seen that companies from the sample started offering a proprietary product and opened the product later in the development process of the product. Nevertheless, not all interviewees agree on the benefits of Open Source and whether offering an OSS product will be sufficient for start-ups to grow. Nevertheless, these companies have grown into successful companies and [12, p.19] confirm that Open Source "makes it possible for small firms to be innovative and find sustainable revenue streams".

The sample data shows us that not all companies from the sample have received investments. Additionally, none of the stakeholders has mentioned investments as a prerequisite for a start-up to become successful. Therefore we disagree with the statement that *OSS businesses rely on investments because they offer free software*. This has also to do with the value exchanges between the different stakeholders in a OSS business model. As [32, p.6] state; "[In OSS businesses] resources are accessed through collaborative relationships between two or more parties". Reviewing the e3-value model based on these relations in Figure 1 we can state that *different value streams are needed to create a successful OSS company*. Moreover, the interview transcripts and the e3-value model show us that *OSS enables a growing customer base* because the customers are not the only users. The developers from the OSS community are often partially working on the product for their own benefit. When the community grows, the customer base is also grows. Additionally it is known that OSS can be offered for a lower price since the software is provided by an outside party [32] and in turn low software prices attract new customers. The OSS contributes in another way, where we can state that *the strategic advantage of OSS is the already existing network bounds offered by the communities*. Interviewees agree with the idea that being Open Source accommodates the establishment of a business and its OSS product. We can conclude that no matter which OSS business model is chosen, a business can choose to make the source code freely available and still serve its business interests as a for-profit organization [33, p.46].

To be able to create a model for entrepreneurs that have the aspiration to start an OSS company, question 10 of the interview aimed at tips given by the experts from the OSS market. According to interviewee F., company F. extracts its revenue from new features in their core products. Nevertheless, the interviewee recommends start-ups to use libraries that are already available to create your own product and to build your own marketing idea around it. Or shortly, take available code from the web and create a special offer. The fact that the software is being used helps in the improvement of the product. On a certain moment in the development phase, it will be time to invest in the core product. This is needed to improve the core layers in your own software product and to keep the whole community rolling. The interviewee has a critical view on start-ups because he sees that some do not "grow up" at a certain point in time, staying for a long time in the initial development phase.

Interviewee of company E. emphasizes client communication, and if you do not have any clients as a start-up: just find one client. This aligns with what interviewee H. says; identifying the right kind of customers and making them happy. Interviewee E adds that entrepreneurs must make sure that a business model covers the costs and provides decent profit margins. Also, engaging the open source community and being honest and transparent with all stakeholders is mentioned by company H.

Interviewee E. has a simple suggestion, stating: *"Just create a product that 1 person thinks is good, a product that 1 person likes and iterate on this product until they get to the point they are very happy with it and want to give you money for it."*. The other option suggested by interviewee E. is to attract venture capital, but they are often not interested in an OSS product. Nevertheless, according to interviewee J. venture capital can be attracted and obtained by giving lots of presentations for possible investors. What the interviewee sees, is that successful companies which originate from the open source world, first created an open source project which became very popular and then started to build a business around it. The interviewee concludes that it might be better to establish a start-up around a proprietary product to earn money and open the software in a later stadium. He states: *" I think that that is a important realization, you can make things open source only one time."* There is the possibility to change the license, but every time you do that the community will not like it. After the company created an enterprise product, the challenge was to price it right. It is easy to make your product very cheap, but company J. never lost any clients by raising the price of their product. Nevertheless, the pricing should be appropriate and according to the quality of the product. Interviewee B. states that, if they look back at the open source product in the initial phase, it was too good which caused the problem that enterprises did not want to make the step to using their paid enterprise edition. Because of their ambition to grow, the company decided to put less focus on the open source version and more on the enterprise software. The interviewee says that in the end it entrepreneurs must focus on what the client currently and in the future wants.

Interviewee J. agrees with the before mentioned idea of creating a product which is interesting for the market. You should differentiate your company from others and the interviewee sees three ways to do so:

- **Comparative differentiators**; The other company has features A, B and C, so we have features A, B and C.
- **Unique differentiators**; We can do what the other company cannot.
- **Holistic differentiators**; These are the aspects like the company culture and how you are perceived in the market. This comes up later in the sales processes after the acceptance of a client to start using the software. These differentiators are less seen in the begin phase of the acceptation because there is a need for awareness on the vision of the company and the changing technology trends.

6 Findings

The Software Business Model frameworks show several correlations between combinations of business model components. First we recapitulate the main findings from the completed SBMFs. In terms of strategy, the OSS companies have quality as their value proposition, which correlates with the fact that "development" is identified most often as the main component in the value chain. The sales volume of the sample differs between medium and high but comes directly from the source. The pricing assessment base is a hybrid combination of usage-based and usage-independent pricing. We recognize a subscription model in the recurring payment flow structure. We see that mostly the companies offer Application software, with Cloud Computing software coming second. The platform on which it is offered is a server and the product is usually standardized in bulk. The license model used on the open source part of the offered product is a Copyleft model. Nevertheless, some of the companies offer both proprietary and OSS simultaneously by applying a dual licensing model. The key cost driver is based on costs spent on Research & Development. Localization is mostly worldwide and the type of customers are typically large organizations (¿250 employees) from varying industries. The target end-users are mostly Business dedicated specialists. The marketing channel is maintained by sales agents. The usage component is composed of the offered services. The operating model is a hybrid combination of on premise and on demand, based on what the customer prefers. Also the support model is a hybrid combination, the choices for customers are standard or customer specific support.

The enterprise editions supplied by the case companies form a contradictory business model category called "commercial open source businesses" [23, 34]. The definition of companies that apply this model is given by [23, p1.]: "Commercial OSS projects are owned by a single firm that derives a direct and significant revenue stream from the software." In the commercial open source business model, commercial OSS companies foremost focus on providing services around the software product [34]. The majority of the sample offers their clients SLAs next to the open source or built in a commercial version. The payment flow structure used by the sample is either recurring or hybrid which is a characterizing feature of a subscription.

Altogether, the SBMF data shows that the sample has corresponding components that form a particular business model. This combination can be molded into a blueprint for a hybrid open source business model (see Figure 2). The companies from the sample can be categorized as commercial open source vendors. This specific combination of business model components is based on the success of the sample and forms the proposed blueprint revealing the building blocks of their open source business model.

Based on the investment horizon component of the framework we can state that the sample is looking for growth in company size, number of customers, and community size. This disagrees with the findings of [12] who states that Open Source is a production paradigm that does not support company growth. Since the sample consists of mostly medium and large companies we can state that these companies have grown since the start-up phase while offering an OSS product.

For practitioners, we extracted the following advice from the case studies:

Strategy				
Value proposition: Quality	**Investment horizon:** Growth model	**Value Chain:** Development	**Degree of vertical integration:** Medium	**# of cooperation partners:** Few
Revenue				
Sales volume : Medium High	**Revenue source:** Direct	**Pricing assessment base:** Hybrid combination	**Payment flow structure:** Recurring	**Revenue distribution model:** Medium
Upstream				
Software stack layer: Application software	**Platform:** Servers	**License model:** Open source - Copyleft	**Degree of standardisation:** Bulk production	**Key cost driver:** Research & Development
Downstream				
Localization: All	**Target customer:** Large organisations	**Target industry:** All	**Target user:** Business-dedicated specialists	**Channel:** Sales agents
Usage				
Implementation effort: Medium	**Operating model:** Hybrid combination	**Maintenance model:** Monthly Yearly	**Support model:** Hybrid combination	**Replacement strategy:** Few releases

Fig. 2. A OSS business model blueprint.

- Find a good balance between business oriented employees and open source developers.
- Use the open source community to find developers that can work in-house on your product.
- Funding is not necessary but creates possibilities, otherwise look for incubators.
- Create a clear vision for your company and establish your differentiators.
- Stick to the open source idea, this will keep the community close to the company.
- When the community does not kick-off, start with offering proprietary software and make it open source after a while.
- Communicate frequently with your clients to establish their needs.
- Decide your revenue stream: Services, support, dual licensing, enterprise editions etc.
- Medium and large customers are willing to pay large amounts of money for services, so do not sell yourself short by offering services for low prices.

We suggest the following adjustments to the Schief framework to make it up-to-date and applicable for more specific types of business models.

- **Sales channel**; The interviewees agreed on having sales agents who accommodate the sales and marketing department, but additionally mention their use of online advertising to market their products. Details on the type of the (online) advertisements could give an indication of the revenue creation initiated by advertising.
- **Support model**; details around the support model could be specified by looking at the specific price modules, payment flow structure, if it is based on a stand-alone subscription model or part of a large set of services in- or excluding the product. This is also interesting for the construction of their revenue stream, where we could review how much of the revenue is coming from services versus from the software product itself.

7 Conclusion

In this study we examined and compared the business models of 10 B2B OSS companies. Business models have proven to be useful as a conceptual tool to analyze the revenue logic. The comparison was established by using the Software Business Framework of [11] for the interview protocol and analysis of the business models. The information provided by the interviewees has showed us that it is not harder to establish a business around OSS than around a proprietary product.

We can place the applicability of this research in the field of OSS software, entrepreneurship, OSS businesses and strategy. The blueprint and guideline together are useful for entrepreneurs who want to start a business around an existing open source project, or for who want to change their business model when growth is not part of the long-term picture anymore. Furthermore, investors can compare the business model of prospective start-ups they want to invest. Overall, this research provides an overview of the current used business models in the OSS market and mentioned in literature. This overview is offered in the shape of a blueprint for current OSS practitioners, start-ups and researchers.

In this case study, the use of the SBMF could threaten the internal validity because of the interviewees' lack of knowledge in the use of the framework. Threats to the construct validity can be found in the extent to which the experiment setting, in this case B2B OSS businesses, reflects the construct under study. The sample could be expanded to B2C companies by applying additional sample criteria. External validation of the blueprint can be expanded by be applying it on the business models of OSS start-ups. To eliminate the threat on external validity, the blueprint can be put into use on businesses that do not have a business model yet, or on businesses that want to change their current business model and strategy. Future research could possibly focus on the influence of the community on the business model and the development of the product. Ultimately, more research on OSS business models should for start-ups is needed. The field of OSS, start-up strategies and investments should be further explored to add to the current body of knowledge.

References

1. Hippel, E.v., Krogh, G.v.: Open source software and the private-collective innovation model: Issues for organization science. Organization science **14**(2) (2003) 209–223
2. Lee, S.Y.T., Kim, H.W., Gupta, S.: Measuring open source software success. Omega **37**(2) (2009) 426–438
3. Raymond, E.S., et al.: Open sources: Voices from the open source revolution. (2000)
4. Fitzgerald, B.: The transformation of open source software. Mis Quarterly (2006) 587–598
5. : The open source definition. (1999)
6. Krishnamurthy, S.: An analysis of open source business models. Making sense of the Bazaar: Perspectives on Free and Open Source Software (2005) 279–296
7. Chesbrough, H.W., Appleyard, M.M.: Open innovation and strategy. California management review **50**(1) (2007) 57–76
8. Lerner, J., Tirole, J.: The open source movement: Key research questions. European economic review **45**(4) (2001) 819–826
9. Androutsellis-Theotokis, S., Spinellis, D., Kechagia, M., Gousios, G., et al.: Open source software: A survey from 10,000 feet. Foundations and Trends® in Technology, Information and Operations Management **4**(3–4) (2011) 187–347
10. Onetti, A., Capobianco, F.: Open source and business model innovation. the funambol case. In: Proceedings of the first International Conference on Open source Systems. (2005) 224–227
11. Schief, M.: Business models in the software industry: the impact on firm and M&A performance. Springer Science & Business Media (2013)
12. Bonaccorsi, A., Giannangeli, S., Rossi, C.: Entry strategies under competing standards: Hybrid business models in the open source software industry. Management Science **52**(7) (2006) 1085–1098
13. Baxter, P., Jack, S.: Qualitative case study methodology: Study design and implementation for novice researchers. The qualitative report **13**(4) (2008) 544–559
14. Yin, R.K.: Case study research: Design and methods (applied social research methods). London and Singapore: Sage (2009)
15. Pervan, G., Maimbo, M.: Designing a case study protocol for application in is research. In: Proceedings of the Ninth Pacific Asia Conference on Information Systems, PACIS (2005) 1281–1292
16. Eisenhardt, K.M.: Building theories from case study research. Academy of management review **14**(4) (1989) 532–550
17. Robinson, O.C.: Sampling in interview-based qualitative research: A theoretical and practical guide. Qualitative Research in Psychology **11**(1) (2014) 25–41
18. Schief, M., Buxmann, P.: Business models in the software industry. In: System Science (HICSS), 2012 45th Hawaii International Conference on, IEEE (2012) 3328–3337
19. Cooper, D.R., Schindler, P.S., Sun, J.: Business research methods. Volume 9. McGraw-Hill Irwin New York (2006)
20. : What is nvivo?
21. Gordijn, J., Akkermans, J.: Value-based requirements engineering: exploring innovative e-commerce ideas. Requirements engineering **8**(2) (2003) 114–134
22. OMahony, S.: 20 nonprofit foundations and their role in community-firm software collaboration. (2005)

23. Riehle, D.: The commercial open source business model. In: Value creation in e-business management. Springer (2009) 18–30
24. Rosenfall, T.: Open Source Vendors Business Models. PhD thesis, Linköping University Electronic Press (2012)
25. Morris, M., Schindehutte, M., Allen, J.: The entrepreneur's business model: toward a unified perspective. Journal of business research **58**(6) (2005) 726–735
26. Hoch, D.J., Roeding, C., Lindner, S.K., Purkert, G.: Secrets of software success. Harvard Business School Press Boston (2000)
27. Stewart, K.J., Ammeter, A.P., Maruping, L.M.: Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects. Information Systems Research **17**(2) (2006) 126–144
28. Onetti, A., Verma, S.: Licensing and business models. Technical report, Department of Economics, University of Insubria (2008)
29. SEC: Securities and exchange commission: Form 10-k. (2013)
30. Buxmann, P., Diefenbach, H., Hess, T.: The software industry: Economic principles, strategies, perspectives. Springer Science & Business Media (2013)
31. Whichmann, T.: Floss final report: Firms' open source activities: motivations and policy implications. (2002)
32. Lindman, J., Rajala, R.: How open source has changed the software industry: Perspectives from open source entrepreneurs. Technology Innovation Management Review **2**(1) (2012)
33. Hecker, F.: Setting up shop: The business of open-source software. IEEE software **16**(1) (1999) 45–51
34. Popp, K.M.: Leveraging open source licenses and open source communities in hybrid commercial open source business models. In: IWSECO@ ICSOB. (2012) 33–40

How to support transformation from on-premise products to SaaS?

Position paper for future research

Teppo Yrjönkoski

Tampere University of Technology, Pori, Finland
teppo.yrjonkoski@nelnastori.fi

Abstract. This paper reviews academic knowledge for software-intensive business firms' approaches to support transition from on-premise solutions to SaaS. The aim is to increase preunderstanding for future research and review the transformation's impact on business models. The study is restricted to the small and medium-sized software vendors. In addition, embedded software vendors are excluded from the research. In preliminary unsystematic literature review, several business model specifications and canvases used to address the transformation were identified. Firstly, a few of the papers were concentrating on huge software-intensive companies like Oracle, Siebel etc. and comparing their business models. Secondly, other studies were analyzing technology changes as well as threats and the lifecycle of technology. Thirdly, researches were analyzing SaaS platforms like (Microsoft's) Azure or (Amazon's) AWS. The review shows that few works focused on how the smaller enterprise software companies did the transfer, which covers for example personnel, product portfolio, distribution network, market segmentation and revenue model or why they have not even started. This study shows that there is lack of studies addressing this issue and propose further research on the issues, which would benefit small- and medium-sized software-intensive firms.

Keywords: Cloud Computing, Software-as-a-Service, business model in software business, from on-premise to Software-as-a-Service, Software-intensive business.

1 Introduction

Cloud computing and Software-as-a-Service (SaaS) paradigm have gained remarkable popularity in the software industry. According to NIST [37] definition, cloud computing refers *to "enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction".* There are several different service models inside the cloud computing paradigm; however, most often used are IaaS (Infrastructure-as-a-Service), PaaS (Platform-as-a-Service), and SaaS (Software-as-a-Service). SaaS refers to "*[…] capability provided to the consumer is to use the provider's applications running on a cloud infrastructure"* [37].

For a customer as well as a software vendor, cloud computing and SaaS solutions offer clear benefits. On one hand, a customer will be using the same software version as everyone else. Consequently, there will be less bugs, less maintenance, faster product development for the customer etc. The negative point is that there are no or only a few alternatives to adjust or tailor the software to fit specific customer needs.

On the other hand, the vendor has just one main version to develop, update and keep updated compared to an on-premise alternative, where there may be two versions under support and one new version under development. The positive impact of these all is the improved speed of product development and cost savings because of less concurrent work. The negative impact is, firstly, minor customer requirement coverage. Secondly, this may need increased attention to product management and product marketing. If there is a lack of those activities, it may lead the customer choosing another vendor.

Due to the rising popularity of SaaS solutions, several software companies have changed their business model from selling on-premise products to providing cloud-based solutions. As a notable example, for instance Microsoft has transformed its Office tools from on-premise installed products towards SaaS-like solutions with the new Office 365 service. In addition, it is likely that there are plenty of companies, which are planning of following the same path.

However, the transformation process from offering an on-premise installed product to a solution offered as a service in a cloud is not straightforward. In top of technical challenges, this kind of a transformation process naturally creates change pressures to the company itself and its business model. For example, how an organization, which previous on-premise software product has generated income with both license and support sales, should manage changes in the cash flow when a new SaaS version generates stable, yet in the beginning smaller, revenue stream?

This position paper focuses to study what do academic literature report of this kind of transformation processes. Our focus is specifically on the changes in the business model as well as in the financing of a software-intensive vendor. The aim of the paper is to create a starting point for further studies in this area as well as propose some lines of research. This position paper uses unstructured literature review method [23] to collect relevant primary studies for the starting point. Based on the findings of the literature review, we discuss on potential areas for future work.

The digital transformation impacts the whole industrial world. Because the digital transformation will be everywhere, there will be a risk open the limitations too much towards generic digital disruption. For this paper, we restrict our attention to software-intensive companies, and especially small and mid-size software vendors. The target group is enterprise software vendors and non-software companies have leaved out. The rationality is that large software vendors might have enough knowledge, capital and resources to manage the transition whereas small- and mid-size enterprises might not have capital required nor enough human resources for a new project. Furthermore, we exclude embedded software vendors as their main revenue flow often does not come from selling software licenses. That is, the transition might not create similar changes to their business model.

The remaining of this paper is organized as follows. Section 2 gives an overview of cloud computing research and defines the complex concept of business models. Section

3 presents finding from the literature review regarding different cloud business models and business model elements. The fourth section discusses about the directions of further research and the final section concludes the study.

2 Background

2.1 Cloud computing

The paradigm shift, from on-premise software solutions toward SaaS solutions, seems to be reality nowadays. The SaaS trend seems to be a *de facto* standard or at least approaching the *de facto* on consumer software solutions. [38] On enterprise business software, the picture is not yet the same as in the consumer software.

Marston [33] pointed out two fundamental classification dimensions approaching to study cloud computing: *i)* business issues, and *ii)* technology issues.

However, as this study focuses on the business issues, the issues belonging into the technical perspective will be excluded for keeping the target clear and tight. Therefore, for example issues belonging in the following areas will be excluded from the research:

1. Software product development;
2. Core technologies like virtualization, multitenancy and web services;
3. Software product modulization, product structures and product modules; and
4. Software development methodologies

The consumer software, like mobile phone software, are today more or less platforms where different vendors are producing their applets. Consumers are paying monthly fee or limited purchase price and at the same time, the software vendor pays a fee to the platform owner.

In business software, above mentioned operation model is similar, for example, like SAP has, called SAP EcoHub an online solution partner marketplace. There a single software vendor has a possibility put their software (applet) for purchase by end user. Remarkable is that is only possible for SAP end users who are running it in SaaS format, not on-premise SAP product owners. Other big actors in the field, like Microsoft and Oracle have similar concepts. Of course, for example, SAP is investing huge amounts of resources to go towards SaaS, but it will take several years until the whole on-premise product is rewritten.

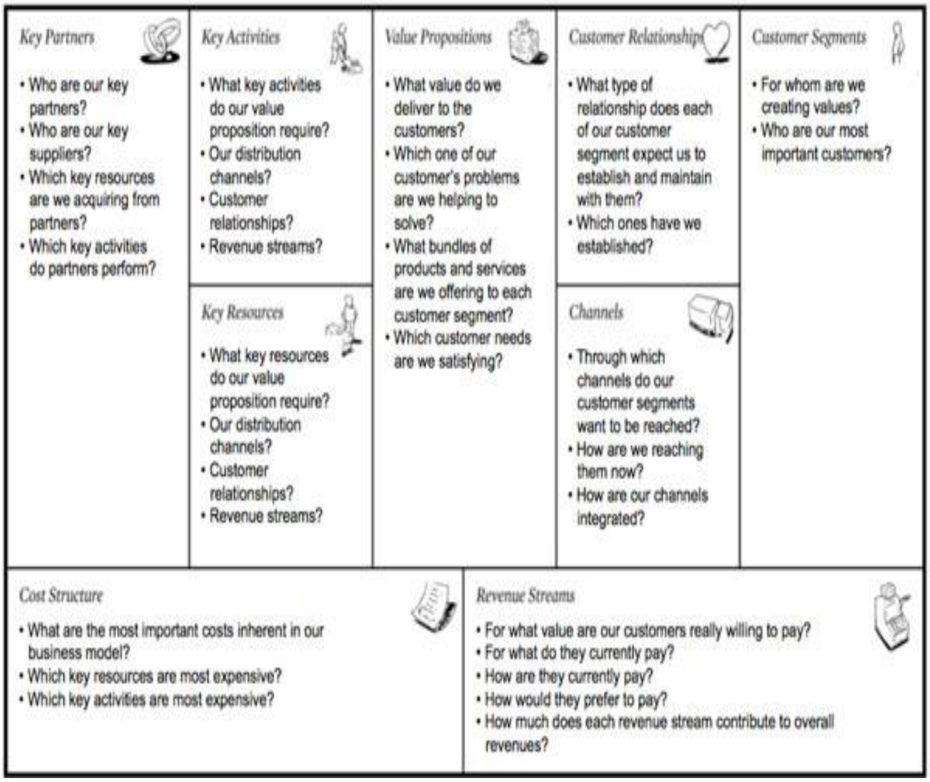
The headache with smaller software firms differs a lot. They have existing product, existing customer base and no platform. The question how jump into SaaS world, might be a question of dead or life. Roughly, based on author's over 20 years' experience of software industry and empirical research, the companies planning to promote a new SaaS product to a market, may be divided into four categories:

1. The first ones have started with something very new without legacy systems headache.
2. The second ones have fight with existing customers with their on-premise installations as well as at the same time try to develop modern cloud-based solutions with new functionalities.

3. The third type of company believes that the momentum is not right to convert the business model because of huge existing cash flow.
4. The fourth have not even started to consider the threats of market change.

In the remaining of this work, we will take a look how academic literature guides the companies belonging into the first two groups.

Fig. 1. Osterwalder and Pigneur's business model canvas [32]



2.2 Business model

A business model is an important concept for this study. As it’s seen in the Figure 1 the business model consists of several factors. Later it will be other descriptions of business model like Table 1 and Table 2. Whenever a business transformation is under discussion, it will always lead to a new business model. On practical level, for example the following question will raise: The firstly what the new revenue model shall be? Secondly, is there a need for a new kind of partnering? Thirdly, what are the products and services in offered portfolio? Fourthly, what are the key resources, do those already exists or will it be the starting point to find right resources first?

Osterwalder and Pigneur's [32] presented a framework for analyzing business model and the changes in business model. Their model is nowadays widely known as the Business Model Canvas. While Osterwalder and Pigneur’s canvas is not the only one, it is the most well-known in both academia as well as industrial world. The canvas is presented in Figure 1. In the Business Model Canvas, there are nine factors which all must be analyzed separately and compared to today's status versus future status. After individual factor analysis, the results should be crosschecked.

Osterwarlder and Pigneur's model is not the only one. Juntunen [21] have analyzed different authors and their opinion of business model elements. Juntunen's summarization of the main work on business models is presented in Table 3. It is noteworthy that there are several different works aiming to define the business model and there are different numbers of components from which a business model has been defined from. Furthermore, most of these works have been published in the during a relatively short time period: during 1998–2002.

Furthermore, Da Silva et al. [14] has characterized business models and its elements in five categories. The elements, which they identified to belong in a business model logic, are presented in Table 2. Da Silva's approach differs somewhat from the other approaches, yet there are common elements such as value proposition and earning logic. It is easily possible to see all shown frameworks for business models vary from each other as well all has its own logic.

Table 1. Business model elements (adapted from [21]).

Authors	Business model elements	Number of elements
Timmers (1998)	Product/service information flow architecture, business actors and roles, actor benefits, revenue sources, and marketing strategy	5
Chesbrough & Rosenbaum (2000)	Value proposition, target markets, internal value chain structure, cost structure and profit model, value network, and competitive strategy	6
Hamel (2001)	Core strategy, strategic resources, value network, and customer Interface	4
Amit & Zott (2001)	Transaction content, transaction structure, and transaction governance	3
Weill & Vitale (2001)	Strategic objectives, value proposition, revenue sources, success factors, channels, core competencies, customer segments, and IT infrastructure	8
Rayport & Jaworski (2001)	Value cluster, market space offering, resource system, and financial model	4
Afuah & Tucci (2001)	Customer value, scope, price revenue, connected activities, implementation, capabilities, and sustainability	8
Dubosson-Torbay, Osterwalder & Pigneur (2002)	Products, customer relationship, infrastructure and network of partners, and financial aspects	4

Table 2. Elements that reflect the business model logic (adapted from [14]).

Element	Logic
Customer value proposition	Understanding and creating products and services that meet customers' needs and help them fulfil their goals.
Earning logic	Designing a revenue model leading towards a sustainable business.
Value network	Designing value-added relationships with partners that represent the extended enterprise of the organization.
Resources and capabilities	Leveraging and repurposing existing or acquiring new resources and capabilities to create products and services of value to customers and generate consequent revenue.
Strategic decisions	Decisions aimed at creating a sustainable competitive advantage.

Luoma [28] pointed out that the determination which IT company is service firm, which is product firm, may be complex. There might be a product firm whose revenue just 20% are license sales and the rest 80% of revenue are services like designing, implementing or operation.

The term product or service company is still unclear and requires deeper research. The most important factor may be is there a common model how those companies behave. Rather often companies have either product or service operations in place. The most of companies have both operations. When investigating the transfer of business model change from on-premise to SaaS it must be sure are people talking about product or service company.

For example, if the company A turnover split is:

- 20% license sales
- 40% consultancy sales
- 40% maintenance

Compared to company B:

- 60% license sales
- 40% services.

The operational business structure will vary remarkable depending the level of product / service allocation

3 Results

Cloud computing start to be common nowadays. A lot of research work has done to justify what is cloud computing. However, the main target in this research is to review what is known on enterprise software firms and how the business model has changed by moving from on-premise to SaaS business model.

For this study, an unstructured literature review [23] was selected as the method. The justification was that the authors were unaware whether there would be enough primary studies for a full-scale systematic literature review. Therefore, a lightweight unsystematic literature review was used to map the current status of the field for further analyses. Based on this unstructured review, a systematic literature review could be implemented later, and the findings of this study can be used as a control group for the review.

The unsystematic literature review was performed so that the authors searched primary articles with different keywords and their combinations. The used keywords included e.g., *cloud computing*, *SaaS*, *business model*, *transformation, change.* The searches were done with, e.g., Google Scholar, IEEE Xplorer, ACM Portal and ScienceDirect publication databases.

Articles which were found relevant for this study was selected and read through. If a primary study referred to another primary study, that was not included into, the other primary was acquired and included into the review. We included also other than research articles (e.g., reviews in magazines) if they were finding to belong in the target group. The final set of selected articles are [1-10, 12-22, 24-30, 35-36].

By doing unstructured literature analysis, it was found that there are several alternative approaches to narrow the business model in this context. One fact was already now rising: The business model will be the most important factor if the transfer will be successful or not.

In the following, we will review the literature what was found in the unstructured review. For example, Boilat and Legner [3] has done a research of Enterprise software and cloud computing. They summarized existing research in a table (c.f. Table 1). Their findings were noticeable: *"From multiple case studies covering traditional and pure cloud providers, we find that moving from on-premise software to cloud services affects all business model components, that is, the customer value proposition, resource base, value configuration, and financial flows"* [3]. However, it is worthy to note that their study did not explicitly focus on how to carry out transformation from an on-promise setting to a SaaS solution. Yet, their findings emphasize the importance of business model in understanding the cloud computing paradigm shift in software-intensive businesses.

In addition, existing research divides SaaS environments into subclasses. One alternative for dividing SaaS solutions is based user involvements as Luoma et al [29] have done. They have found three classifications:

- Enterprise SaaS
- Pure play SaaS
- Self-Service SaaS

All those three classifications they have analyzed by financial, resource-base and customer-facing elements. Boilat and Legner [3] used the same division and classified business model element according to these (c.f. Table 3).

Regarding more general business-oriented research on cloud computing and SaaS, there are many studies. Thus, SaaS has started to be commodity. For example, a widely known model how to analyze different factors in cloud computing is a Cloud Cube Model from the Jericho Forum [20]. It has developed targeting to understand different factors around the cloud operations. Cloud Cube Model is illustrated in Figure 2.

Cloud Cube Model has been further analyzed and developed by Chang [7]. Their specialty was identifying different sorts of business types and strengths and weaknesses of each business types in cloud computing. Chang [7] classified cloud computing business models and found eight business types:

1. Service Provider and Service Orientation,
2. Support and Service contracts,
3. In-House Private Clouds,
4. All-in-One Enterprise Clouds,
5. One-Stop Resources and Services,
6. Government Funding,
7. Venture Capitals, and
8. Entertainment and Social Networking.

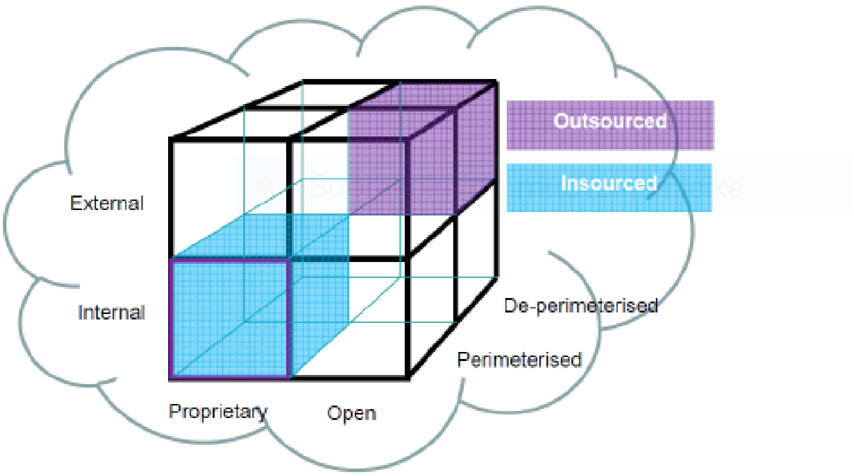


Fig. 2. Cloud Cube Model [20]

Table 3. Existing research on enterprise software and cloud computing (adapted from [3]).

Authors	**Focus**	**Customer perspective**	**Vendor Per-spective**
Benlian et al. (2009)	SaaS adoption by firms	X	
Choudhary (20079	Switch from perpetual software licensing to SaaS and its impact on software quality	(X)	X
Ellahi et al. (2011)	Cloud deployment models, issues of moving enterprise applications to the cloud, and the market evolution for enterprise cloud computing	X	
Janssen & Joha (2011)	SaaS doption in public sectors (ministries, public agencies, municipalities)	X	
Katzan (2009)	Cloud computing from a business and architecture perspective	X	X
Khajeh-Hosseini et al. (2010)	Research challenges for cloud computing from an enterprise or organizational perspective	X	X
Liao (2010)	SaaS business model for enterprise software	X	X
Luoma et al. (2012)	ASP and SaaS firms' business models	X	X
Leimeister et al. (2010)	Actors, roles, and business aspects of cloud	X	(X)
Loebbecke et al. (2012)	Practical case of cloud computing assessment	X	
Mangiuc (2011)	Challenges and risks of moving applications to the cloud	X	
Marston et al. (2010)	Overview of cloud computing; SWOT analysis from a business perspective	X	(X)

Table 4. SaaS solutions classification (adapted from [29]).

Element group	Element	Enterprise SaaS	Pure play SaaS	Self-service SaaS
Customer-facing elements	Value proposition	A mass-customized but complex application that also requires support services	Horizontal, standardized web-native application	A very simple application that is easy to adopt
	Customer segments	Larger enterprises and their IT managers and top executives	SMEs, middle management and end users	Adopted first by end users and individual consumers, then SMEs
	Customer relationship	High-touch, trust-enchanting customer relationships with tailored contracts	Less human contact in deployment required than traditionally, owing a simpler applications	Fully automated self-service; as little interaction with the customer as possible
	Channels	Perform personal sales and employ channel partners	Sales channel is push-oriented, and SaaS firms engage in inbound, high-pressure sales	Outbound and viral marketing used to attract customers to the vendor’s homepage. Landing page critical in turning prospects into customers.
Resource-base and value configuration elements	Key resources and activities	Possess domain expertise and utilize an ecosystem of companies as a resource	Both domain expertise (to include best practices into the application) and application development capabilities	Close to zero marginal costs
	Key partners	User partners to deliver value-adding applications and services	IT service providers for infrastructure and support services	N/A
Financial elements	Revenue streams	Vendors charge an entry fee, recurring fees, and services fees	Small entry fee and a recurring fee	Use of freemium model, ad-based revenues or small recurring fees
	Cost structures	Have varying marginal costs, owing to the long sales cycles and required support	Initial development costs may be high, but firms aim for minimal marginal costs	N/A

In a business model transformation, personnel are one of the most critical factors. In computing world and all high-tech industry, there is huge lack of competent people [39]. A business model change to adopt into the requirements of a modern business world is necessary for a company, eventually. At least in Scandinavian, software-intensive firms are not able to change all resources and at same time and start a new product development project with fresh resources. Personnel is a big part of success. Thus far, only Sultan [36] addressed organizational culture in a cloud computing setting. Yet, their focus is on the organizations and their culture, not guiding how to manage transformation as a software-intensive firm.

Transformation from on-premise to SaaS moves the business logic from product business to service business. Cusumano's recent work [12, 13] covers that area; however, he does not give practical guidelines for companies, but instead focus on market-level discussion. Da Silva [14] has analyzed the impact of disruptive technologies to business model comparing Siebel and Sales Force as well as Amazon and Sales Force. It is worthy to note that those companies are huge compared to target firms in this position paper.

Finally, Juntunen [21] has looked the transformation issue by using dynamic capability view and Chesbourgh [9] is more concentrating on innovations in business model. Marston [30] has a business perspective approach for the subject. However, also these studies do not focus on giving practical guidelines for a software-intensive firm.

4 Discussion

The aim of this position paper was to review the current knowledge of academic literature on guiding small and medium-sized software-intensive businesses for transforming their business model from on-premise products to SaaS solutions. In the unstructured literature research, it was found that there are several investigations and research results comparing companies, their status in cloud development and their product portfolios. Mainly the studies in the extant literature have been focused huge companies like Oracle and SAP.

However, there seems to be lack of research to comparing companies how they have done the technology and business model transformation from on-premise to SaaS business. Specifically, there is a lack of studies how smaller firms have achieved the goal. While it is possible that there is such research available; however, there are lack of understanding to support the companies in this kind of transformation and this requires do more detailed research.

Thus, this position paper requires further research concentrating on small and midsize software companies, who are on their way to transfer their on-premise product range to SaaS software. The main goal of this kind of research should be to answer to the following questions:

- How software-intensive firms have handled the transformation and what has been the lessons learned?
- What are the required steps in transformation?
- What has been the critical factors in business model transformation?

- What guidelines could research give to companies that are planning of transforming their product offering and business model?
- How a software-intensive business can satisfy simultaneously both its current customer base, with on-premise installations, as well as the new customers, with wishes for new functionalities in a SaaS solution?

Based on the unstructured literature analysis there are few main limitations that should be acknowledged in the research. Firstly, what is the impact of product / service allocation in business model and business model transformation for a software-intensive business? Secondly, what is the impact of company size for this kind of a transformation? Thirdly, what is the impact of life cycle status, is the company well established or a start up, to the transformation?

The continuation of this literature research will be to find out candidate companies and then for example, analyze their cloud computing business models and do the classification like Chang [7].

5 Conclusion

This study focused on searching what the extant knowledge reports on transforming a business model of a software-intensive business from an on-premise product to a SaaS solution. Considering the researched material, there were several studies reporting differences caused by an adaptation of a cloud computing-bases business model. However, most of the review work focused on large-sized companies, which have resources to manage the transformation. On the contrary, there are not much reported on small and medium-sized companies.

Limitations of the paper are lacking systematic literature review and other methodology. Ecosystems business model should need more attention timely and rigour academic literature.

With the studied reference literature, this study has shown that there is a need to research how to support a small or midsize enterprise software company, which is planning to change the business model from on-premise to a SaaS business model.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. O. N., & Zaharia, M. (2010). A View of Cloud Computing. Communications of the ACM, 53, 50-58.
2. Bensinger, G. (2012). Competing With Amazon on Amazon. Wall Street Journal. Retrieved from http://online.wsj.com/article/SB10001424052702304441404577482902055882264.html
3. Boillat, T., Legner. C. From On-Premise Software to Cloud Services: The Impact of Cloud Computing on Enterprise Software Vendors' Business Models
4. Boulton, C. (2012). Oracle Customers Rankled by Product Roadmap. WSJ Blogs - The CIO Report. Retrieved from http://blogs.wsj.com/cio/2012/04/02/oracle-customers-growing-angrier/

5. Braganza, A., Awazu, Y., & Desouza, K. C. (2009). Sustaining innovation is challenge for incumbents. Research-Technology Management, 52(4), 46–56.
6. Casadesus-Masanell, R., & Ricart, J. E. (2011). How to design a winning business model. Harvard Business Review, 89(1/2), 100–107.
7. Ghang,V, Bacialupo,D, Wills, G. De Roure, D. A Categorisation of Cloud Computing Business Models
8. Chesbrough, H. Business Model Innovation: It's not just about Technology Anymore, Strategy & Leadership, vol. 35, no. 6, pp. 12-17, 2007.
9. Chesbrough, Henry. (2010). Business Model Innovation: Opportunities and Barriers. Long Range Planning, 43(2–3), 354–363. doi:10.1016/j.lrp.2009.07.010
10. Choudhary, V. Software as a Service: Implications for Investment in Software Development, in Proceedings of the 40th Annual Hawaii International Conference on System Sciences, Waikoloa, 2007, pp. 209a
11. Christensen, C. M. (1997). The innovator's dilemma: when new technologies cause great firms to fail. Harvard Business Press.
12. Cusumano, M. The Changing Software Business: Moving from Products to Services, Computer, vol. 41, no. 1, pp. 20-27, 2008.
13. Cusumano, M. (2010). Cloud computing and SaaS as new computing platforms. Communications of the ACM, 53(4), 27–29.
14. DaSilva, C.M., Trkman, P., Desouza, K., Lindič, J. Disruptive Technologies: A Business Model Perspective on Cloud Computing Technology Analysis & Strategic Management, 2013
15. A. Dubey, D. Wagle, Delivering software as a service, The McKinsey Quarterly (May 2007) 1–12.
16. Ellahi, T., Hudzia, B., Li, H., Lindner, M.A., Robinson, P. The Enterprise Cloud Computing Paradigm. USA: John Wiley and Sons, 2011.
17. Gartner. (2012, November) Gartner: Top 10 Key Technology Trends for 2013. CloudTimes. [Online]. Available:http://cloudtimes.org/2012/11/06/gartner-top-10-key-technology-trends-for-2013/.
18. Hugos, M. H., & Hulitzky, D. (2010). Business in the Cloud: What Every Business Needs to Know About Cloud Computing (1st ed.). Wiley.
19. Irwin, S. (2012). Enterprise 2.0: Freemium first, enterprise second (Part 1 of 3). GigaOM. Retrieved October 30, 2012, from http://gigaom.com/2012/04/28/enterprise-2-0-freemium-first-enterprise-second-part-1-of-3/
20. Jerico Forum "Cloud Cube Model: Selecting Cloud Formations for Secure Collaboration Version 1.0", Jerico Forum Specification, April 2009
21. Juntunen, M., Business model change as a dynamic capability. Doctoral thesis. University of Oulu 2017
22. Kim, W. C., & Mauborgne, R. (2005). Blue Ocean Strategy: How to Create Un-contested Market Space and Make Competition Irrelevant (1st ed.). Boston: Harvard Business Press.
23. Kitchenham,B., Charters,S., (2007): Guidelines for Performing Systematic Literature Reviews in Software Engineering. Version 2.3, Technical Report, Software Engineering Group, Keele University and Department of Computer Science, University of Durham.
24. Leimeister, S., Riedl, C., Bõhm, M., Krcmar, H. The Business Perspective of Cloud Computing: Actors, Roles, and Value Networks in Proceedings of the Eu-ropean Conference on Information Systems 2010, Pretoria, 2010.
25. Lin, A., & Chen, N.-C. (2012). Cloud computing as an innovation: Percepetion, attitude, and adoption. International Journal of Information Management, (0). doi:10.1016/j.ijinfomgt.2012.04.001

26. Low, C., Chen, Y., & Wu, M. (2011). Understanding the determinants of cloud computing adoption. Industrial Management & Data Systems, 111(7), 1006–1023.
27. Luoma, E., Rönkkö, M., Tyrväinen, P. Current Software-as-a-Service Business Models: Evidence from Finland, Software Business, vol. 114, no. 2, pp. 181-194, 2012.
28. Luoma, E., Examining Business Models of Software-as-a-Service Companies. Doctoral Thesis University of Jyväskylä 2013.
29. Mahowald, R.P., Konary, A., & Sullivan C.G. (2011). Market Analysis Perspective: Worldwide Saas & Cloud Services, 2011: New Models for Delivering Software. http://www.idc.com/getdoc.jsp?containerId=232239.
30. Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. (2010). Cloud computing-The business perspective. Decision Support Systems.
31. Mayer, M. K. Future trends in model management systems: parallel and distrib-uted extensions, Decision Support Systems 22 (4) (1998) 325–335.
32. Osterwalder, A., Pigneur, Y. Business Model Generation: A Handbook for Vision-aries, Game Changers, and Challengers. New Jersey: Wiley, 2010.
33. Rahikkala, J., Hyrynsalmi, S., Leppänen, V., Porres, I. The Role of Organisational Phenomena in Software Cost Estimation: A Case Study of Supporting and Hindering Factors. E-Informatica Software Engineering Journal, Volume 12, 2018, pages 167-198, DOI 10.5277/e-Inf180101
34. Robinson, D. K. R., Le Masson, P., & Weil, B. (2012). Waiting games: innovation impasses in situations of high uncertainty. Technology Analysis & Strategic Management, 24(6), 543–547. doi:10.1080/09537325.2012.693661
35. Rymer, J. R., Staten, J., Wang, C. (2012, May) Achieve Cloud Economics for Op-erations and Services, Forrester Research. [Online]. Available: http://www.forrester.com/Achieve+Cloud+Economics+For+Operations+And+Services/fulltext/-/E-RES61602.
36. Sultan, N., & van de Bunt-Kokhuis, S. (2012). Organisational culture and cloud computing: coping with a disruptive innovation. Technology Analysis & Strategic Management, 24(2), 167–179. doi:10.1080/09537325.2012.647644
37. Mell, P. & Grance, T. (2013) The NIST Definition of Cloud Computing. NIST Special Publication 800-145. National Institute of Standards and Technology. U.S. Department of Commerce.
38. Buxmann, P., Diefenbach, H. & Hess, T. (2013) The Software Industry: Economic Principles, Strategies, Perspectives. Springer: Berlin.
39. Hyrynsalmi, S.M., Rantanen, M.M, & Hyrynsalmi, S. (2018) Do we have what is needed to change everything? HCC13 2018. IFIP Advances in Information and Communication Technology, vol 537. pp. 111-122. Springer, Cham.

ISO 16355 in Software-Intensive Business

Felix Schönhofen[1], Sixten Schockert[1] and Georg Herzwurm[1, 2]

[1] University of Stuttgart, Chair of Information Systems II (Software-Intensive Business), Stuttgart, Germany
[2] Graduate School of Excellence advanced Manufacturing Engineering (GSaME), Stuttgart, Germany
felix.schoenhofen@bwi.uni-stuttgart.de

Abstract: The Software-Intensive Business represents a shift from value creation in development of new products to a trend towards cross-industry enterprise networks and collaboration, including almost every industry. This also has an impact on the entire development process of products and services as it will be even more important to understand the real customer's demand. ISO 16355 is a standard based on Quality Function Deployment (QFD) for converting fuzzy customer needs into more-specific functional requirements. This research-in-progress paper examines the question of how ISO 16355 can be potentially used to support the development of new Software-Intensive Products and Business models.

Keywords: Software-Intensive Business, ISO 16355, Software-Product-Service Systems, Business Models.

1 Software-Intensive Business

Almost across all industries, the Software-Intensive Business represents a shift from value creation in development, production and marketing of monolithic products to a trend towards cross-industry enterprise networks and collaboration [1]. So-called Cyber-Physical-Systems open the door to the digital world, originally reserved mainly for pure software companies, even for manufacturing industry producing goods that are more tangible. Digital and analogue markets converge. The atomization of products and services, e.g. as bundled microservices instead of apps, raises the number of products and services and their providers. Thus, the strategy of offering only single products for more or less one or few nearly homogenous types of customers is outdated and replaced by hybrid product bundles offered on multi-sided markets. Platforms enable cooperation (i.e. development and sales) of the value creation partners. The digitization is leading to a sustainable change towards a common platform economy offering a huge potential for innovative business models, creating and satisfying customer needs for business success.

For this new field of research a group of international selected researchers established during a Dagstuhl seminar in May 2018 a new scientific discipline called Software-Intensive Business (SIB) [2, 3]. SIB studies organizational arrangements within and between organizations in conjunction with methods and tools for value creation, capture, and delivery based on digital products and services [2]. By analyzing the Dagstuhl report, several challenges in Software-Intensive Business can be identified. We have arranged them in Table 1 according to their main areas they affect within SIB: the technical system (i.e. the cyber-physical system itself), the human/personnel side and the ecosystem connecting different partners.

Table 1: Challenges in Software-Intensive Business [2]

Cyber-Physical System	Human System	Ecosystem
Rapid development [e.g. 4]	Lack of knowledge of important stakeholders [e.g. 4]	Partners may be unwilling to change [e.g. 4]
Time pressure [e.g. 5]	Disruptive innovation is unpredictable [e.g. 4]	Subsidizing one side of the market [e.g. 4]
Data ownership [e.g. 4]	Need for mind shift [e.g. 4]	Increased need for coordination [added from 7]
System integration [e.g. 6]		

From an economical point of view, companies try to take an advantage of this changed situation by developing new disruptive products and services based on these new possibilities. However, a product or product-service-bundle can only be effective if it is able to serve real customer needs. Moreover, in order to guide the companies' efforts towards a constantly evolving and sustainable business, it is essential to really understand the potential customer needs. Not until this learning process is well established, companies are able to benefit from the new solution space in an innovative and promising way [8].

2 ISO 16355

The ISO 16355 offers a kit of methods and tools to assure customer or stakeholder satisfaction by identifying their most important needs. It represents a quality approach, whose main purpose is to establish a defined and repeatable product development process, based on definable targets, the involvement of all relevant customers and stakeholders and the focus on their real needs. ISO 16355 can be used independently of the domain or industry and has been successfully applied for hardware, service and software development [9]. As cyber-physical systems usually combine all three of these aspects as Industrial Software-Product-Service bundles [10], ISO 16355 has the potential to offer a toolset for the SIB domain. Thus, this paper poses the question of how ISO 16355 can be used to support the development of new Software-Intensive Products and Business Models.

ISO 16355 is based on a set of six main principles [9]:

a) Prioritize information to focus
b) Understand how to cause good quality
c) Listen to the voice of the customer
d) Observe the customers situation
e) Capture information from other sources
f) Improve internal communications through the transformation of information between perspectives

Obviously, these six main principles address several different aspects. The principles b)-e) seem to be distinct approaches to elicit the real customer needs. You can elaborate them by listening, observing, analyzing and/or considering preferably all possible sources. Accordingly, we summarize them as “Focusing on customer/stakeholder needs”. The ISO 16355 recommends tools like User Personae and Gemba visits in this area [9].

Principle a) is about a more methodical approach to handle the captured information. Customers pay the most for solutions to their most important problems. Thus, it is essential to prioritize the information obtained from the customer/stakeholders according to customer value. We deduce the second cluster “Prioritization” using e.g. tools like the Analytic Hierarchy Process and the Kano model [9].

Lastly, we have to take a closer look on principle f): Compared with the clusters we concluded before, this principle deals not (only) with the contact to the external stakeholders like the customers or partners. It is about the internal communications in particular. There is a gaining in importance about this factor, especially in the context of Software-Product-Service systems in Software-Intensive Business: There is a need to synchronize all development areas and teams and – moreover – to generate mutual understanding to improve the collaboration. So, the third cluster we worked out is “Collaboration”. ISO 16355 suggests the joint building of cause-and-effect diagrams and L-matrices like the House of Quality in this area

Figure 1 gives an overview of the identified ISO 16355 clusters adding some examples of tools mentioned in ISO 16355.

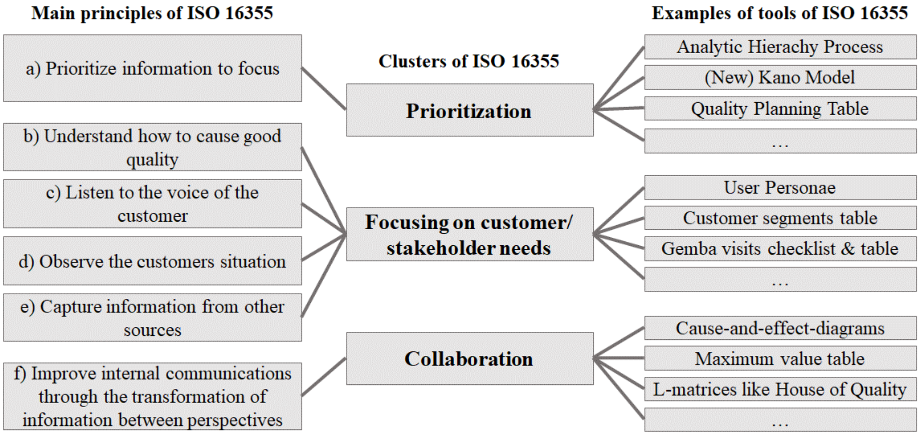


Figure 1: Clustering of ISO 16355 according to main principles of ISO 16355

3 ISO 16355 in SIB

In order to evaluate the general fit of ISO 16355 with Software-Intensive Business, we compare the ISO 16355 - clusters composed of its main principles and corresponding tools (Figure 1) with the challenges in Software-Intensive Business extracted from the Dagstuhl report (Table 1).

Regarding the first two entries in the first column, one have to consider concepts like the Minimal Viable Product: In case there are little resources regarding time and capacity, you have to focus on the really important requirements to satisfy the early users [11]. Apart from the general need to know the customer needs, it is essential to understand which of them are the very most important ones regarding the customer satisfaction and – based on this – to prioritize these top needs.

Regarding data ownership the actors have to find a compromise between privacy and need for data to solve the customers' problem. Up to now, it is not possible to give a "perfect answer" to this challenge, but at least you can say that a lot of communication and collaboration could help to generate a trustful environment. This positive environment should also be helpful to build a common sense of all development areas in the CPS to make system integration smoother. Eventually, prioritization and collaboration seem to be the most significant principles regarding the area of cyber-physical systems.

The lack of knowledge is critical concerning the human aspects within SIB. The mutual understanding of engineering and business benefit from communication and collaboration. In a perceived uncertain and disruptive environment, the focus on the solution-independent real customer needs can solve as a landmark. Nevertheless, enterprises have to develop a positive attitude towards making change, not seldom even requiring a bigger mind shift. Thus again, there is a need for intense collaboration and communication.

This positive attitude towards change is also required in the Ecosystem area: potentially unwilling partners in the ecosystem have to be persuaded. An ecosystem-wide change management, based on smoothly collaboration can serve as a solution. When attempting to establish an ecosystem, it is important to subsidize one side of the market. To accomplish this effectively, it is necessary to understand the needs of the corresponding market side.

Finally, one main problem field in the ecosystem research is the increased need for coordination. ISO 16355's cluster of collaboration is obviously needed for that. Table 2 gives an overview of the SIB challenges and the corresponding main clusters of ISO 16355.

Table 2: Fit of clusters of ISO 16355 to Challenges in Software-Intensive Business (SIB)

System	Challenges in SIB	Main clusters of ISO 16355
CPS	Rapid development	Prioritization
CPS	Time pressure	Prioritization
CPS	Data ownership	Collaboration
CPS	System integration	Collaboration
Human system	Lack of knowledge of important stakeholders	Collaboration
Human system	Disruptive innovation is unpredictable	Focusing on Customer/Stakeholder needs
Human system	Need for mind shift	Collaboration
Ecosystem	Partners may be unwilling to change	Collaboration
Ecosystem	Subsidizing one side of the market	Focusing on Customer/Stakeholder needs
Ecosystem	Increased need for coordination	Collaboration

4 Conclusion and outlook

As shown in this paper, the ISO 16355 and the field of Software-Intensive business share very much the same spirit. The main challenges in SIB and the main clusters of ISO 16355 overlap quite well.

However, as an ongoing research project, this paper represents only the starting point of a more in-depth analysis regarding the application of ISO 16355 in Software-Intensive Business. Most likely, the tools of ISO 16355 have to be enhanced and tailored to cope with the identified challenges in Software-Intensive Business. Due to its high importance within SIB, especially the collaboration side seems to need more in-depth analysis.

In order to develop an approach within the paradigm of customer centricity [12] to create innovative Software-Product-Service systems in the Software-Intensive Business, the identified clusters of ISO 16355 have to be concretized. The result could be a conceptual framework of ISO 16355 for Software-Intensive Business consisting of three levels: the main principles/clusters, the conceptual/methodological approach (e.g. focusing on dealing with fuzzy development tasks in an incremental procedure), and detailed tools/instruments (e.g. incorporating tools for creativity management) [13].

References

1. Acatech (eds).: Cyber-Physical Systems. Innovationsmotoren für Mobilität, Gesundheit, Energie und Produktion. Heidelberg (2011).
2. Brinkkemper, S., Abrahamsson, P., Maedche, A., Bosch, J. (eds.): Software Business, Platforms, and Ecosystems: Fundamentals of Software Production Research. In: Dagstuhl Reports, 1, 1–31 (to be published).
3. Herzwurm, G.: Research approach and strategy – Research of the Chair of Information Systems II of the University of Stuttgart. https://www.bwi.uni-stuttgart.de/en/dept8/researchapproach-strategy/index.html, last accessed 2018/10/05.
4. Bosch, J.: Transforming to a Software Business, In: Dagstuhl Reports, 1, 5–6 (to be published).
5. Bussler, C.: Innovation + Velocity + Pivoting in software production: The new normal. In: Dagstuhl Reports, 1, 6–7 (to be published).
6. Sztipanovits, J., Koutsoukos, X., Karsai, G., Kottenstette, N., Antsaklis, P., Gupta, V., Goodwine, B., Baras, J. und Wang, S. In: Proceedings of the IEEE, 1(100), 29–44 (2012).
7. Mikusz, M.: Cyber-Physical Systems as Service Systems: Implications for S-D Logic. In: The 2015 Naples Forum on Service, 1–19 (2015).
8. Schönhofen, F., Petrik, D., Schockert, S., Herzwurm, G.: Analysis of platform-based Business Models using Quality Function Deployment. In: Proceedings of the 23rd International QFD Symposium, ISQFD, Tokyo (2017).
9. ISO 16355-1: Application of statistical and related methods ti new technology and product development process – Part 1: General Principles and Perspectives of Quality Function Deployment (QFD). ISO 16355-1:2015, Genf (2015).
10. Mikusz, M.: Towards an Understanding of Cyber-Physical Systems as Industrial Software-Product-Service Systems. In: Procedia CIRP, 16, 385-289 (2014).
11. Patton, J.: User Story Mapping: Discover the whole Story. Build the right product. O'Reilly and Associates, Sabastopol (2014).
12. Herzwurm, G.: Position Statement. In: Dagstuhl Reports, 1, 12 (to be published).
13. Schönhofen, F., Schockert, S., Herzwurm, G.: Towards a QFD approach for Software-Intensive Business. In: Proceedings of the 24th International Symposium on Quality Function Deployment ISQFD 2018, 144-149, Hong Kong (2018).

Decision-making in Software Product Management: Identifying Research Directions from Practice

Andrey Saltan[1,2], Slinger Jansen[3] and Kari Smolander[1]

[1] Lappeenranta University of Technology, Lappeenranta, Finland
[2] National Research University Higher School of Economics, Moscow, Russia.
[3] Utrecht University, Utrecht, The Netherlands

{andrei.saltan, kari.smolander}@lut.fi, slinger.jansen@uu.nl

Abstract. Previous studies on software product management (SPM) domain have provided an extensive picture of the work of a software product manager. Still, little evidence exists about what principles should guide their decisions. A product manager's decision-making has a certain level of subjectivity based on managerial intuition. However, sustainable software product development requires effective long-term decision-making practices. Requirements engineering, as well as release planning and roadmapping, are SPM areas with the highest level of evidence-based decision-making. Still, the clear understanding of evidence-based decision-making practices is missing. The paper provides an analysis of decision-making related to SPM, reveals a spectrum of attitudes and approaches and reports assumptions on whether SPM is based on intuition or if it is evidence-based.

Keywords: Software Product Management, Decision-making, Evidence-based management, Case Study

1 Introduction

Software product management is a growing area of research and practice that bridges the gap between business and engineering aspects of the software business. Many frameworks for SPM in diverse areas have been introduced by both researchers and practitioners [1–5]. A systematic analysis of the frameworks has produced core areas of SPM responsibility and activities [6]. Although product management practices may vary significantly across companies and be determined by many internal and external factors, existing research reveals a range of possible SPM practices and provide quite clear picture of the software product manager role. Still, little evidence exists about how the work of software product managers should be organized and what principles should guide their decisions.

Software developing organizations should establish business processes and practices that enable managers to make decisions using evidence rather than intuition. However, the introduction of evidence-based techniques that lead to informed decision-making can be challenging. Especially start-ups in their early maturity stages make decisions with an ad hoc “gut feeling” approach rather than using evidence-based methods [7]. However, a long-term sustainable software product development requires a transition towards more systematic evidence-driven managerial processes and practices [8].

Requirements engineering along with roadmapping and release planning are the areas from which evidence-driven managerial processes and practices begin. However, it is quite common practice to limit only to these areas when companies transform their decision-making practices and process in SPM. Research on evidence-based SPM has also noted this. While recently some studies have raised the question of making SPM more data and model-driven [9–14], we still lack a comprehensive analysis of evidence-driven decision-making and its potential in software product management.

The objective of this paper is to evaluate the feasibility of evidence-based decision-making in software product management as well as to draft a proposal for implementing evidence-based SPM processes, practices and required IT infrastructure. This paper is the very first step towards our research goal to obtain a better understanding of approaches, evidence, and techniques used in SPM decision-making. We also examine product managers’ perceptions on using evidence-based decision-making and identify obstacles of broader implementation of this approach.

2 Background

2.1 Decision-making in SPM

Recent studies indicate the progress of SPM as a discipline at the intersection of software engineering and business domains as well as growing attention from practitioners. Still, multiple challenges for companies can be identified that require proper support from the research community [3, 15]. Overwhelming processes, unclear responsibilities of software product managers along with premature decision-making practices and moving targets are often named as prime challenges [15, 16]. Existing studies provide a solid foundation for roles and scope of duties in SPM [6]. However, SPM efficiency and how SPM decision-making processes and practices should be organized have not received much attention in academic literature.

The range of SPM decisions varies across several dimensions. The decisions can be done in three levels: strategic, tactical and operational levels. At the strategic level, decisions are mostly related to designing a product strategy and defining the overall business model. Decisions at the tactical level intend to guide most product management processes including release planning, lifecycle management, and roadmapping. Finally, decisions at the operational level determine the use of technical solutions for product architecture, required infrastructure and deployment as well as schedules operations. However, like other types of managerial decision, problems of SPM decisions are not limited to these three levels. Factors such as whether the decision is made by a group or individual [17] as well as whether it can be programmed or not [18] may affect decision-making processes and their designs.

Effective decision-making in SPM requires considering a large number of factors. These factors have both engineering and business origins and include market characteristics (e.g., concentration, B2B vs. B2C market type), product specifications (e.g., mass-market vs. bespoke, product vs. digital service), technological (e.g., technology obsolescence, infrastructure, project complexity), organizational (e.g., lack of staff with the required skills, uncooperative internal parties) as well as the customers' needs and expectations (e.g., costs, benefits) [19]. Considering these factors requires collecting vast amount of data and analyzing it with the help of sophisticated techniques and models.

Another attribute of effective decision-making is formalized processes [20]. Using systematic decision-making approaches in SPM creates valuable transparency that in the long run allow all stakeholders to have both clear vision of the situation and to identify a possible room for improvement in vision execution. However, quite often software product managers prefer making decisions in an ad-hoc way driven by their "gut feeling" and to use tacit knowledge, fearing that otherwise their flexibility in decision-making will be reduced [21].

The transition towards SaaS business and development model along with the active use of information management tools that support digitalizing business processes increased the scope and scale of data available for analysis and decision-making in all areas of software product management [22, 23]. However, making sense of this data considering its overwhelming amount and complexity is not trivial. It requires established processes and models for data collection, storage, and analysis with further visualization and integration into existing business landscape and decision-making practices. Without it, many companies still have suboptimal reporting and poor market insight. The inability of obtaining sufficient evidence in the form of data, knowledge or models for informed decisions can push product managers to rely only on the intuitive vision of the product and its life cycle.

For start-up companies, the question of designing decision-making processes and practices with both managerial and technological aspects of new product development has already been raised by Eric Ries [24] and Steve Blank [25]. They proposed "Customer Development Model" and "Build-Measure-Learn" concept that are widely adopted by start-up companies and form the basis for other decision-support solutions for software companies (e.g. The Early Stage Software Startup Development Model [26]). However, the question of how decision-making should evolve through companies' growth and development and what are the prime challenges and trade-offs of decision-making have not received too much attention in academic literature.

2.2 Evidence-based management

Evidence-based decision-making has its origins in evidence-based management that can be defined as practices of "making decisions through the conscientious, explicit, and judicious use of four sources of information: practitioner expertise and judgment, evidence from the local context, a critical evaluation of the best available research evidence, and the perspectives of those people who might be affected by the decision" [27]. Being more an umbrella term rather than a rigorous decision-making approach, evidence-based management does not neglect intuition as a valuable source of expertise. Instead, it assumes that for effective managerial decision-making this intuition should be formalized in the form of verifiable knowledge and supplemented by data-driven and model-driven business analytics as well as consideration of prior experience and conducted researches [28].

Evidence-based decision-making gains a significant boost with the new opportunities to collect, store and analyze data. A new frontier in data management appeared as the "Big Data" concept [29] twisted the overall focus of evidence-based management. The key challenges in the early days of evidence-based management were lack of data that could be turned into pieces of evidence and tweaking the techniques that could help to get at least some proxy data. Nowadays, companies usually have a lot of data, but turning this available "Big Data" into "Smart Data" that could serve as pieces of evidence is not trivial [30]. While recent business and economic studies showed overall rapid adoption of evidence-based management with significant positive impact on company performance [31, 32], defining these practices within the context of a particular company or even an industry could be quite a challenging task.

Evidence-based management has a strong connection with knowledge management as a discipline that intends to manage the processes of creating, organizing, and using the information and knowledge within an organization [33]. Robust knowledge management processes are crucial for effective evidence-based decision-making as it allows to formalize and integrate managers' experience and expertise in decision-making. Moreover, quite often the product management activities are distributed among a group of managers, and each of them has her area of expertise with "tacit" knowledge regarding aspects of product management they are responsible for. In this case, effective knowledge management means storing and sharing this knowledge adequately to ensure informed and coordinated decision-making [34].

The evolution of big data analytics and knowledge management have given a new way of exploring new frontiers in decision making in high volume, front-line decisions. These frontiers are associated with new types and sources of data available, as well as new approaches and methods of analysis to identify evidence required for decision-making [35]. Software companies are the drivers of this process, providing its customers on the B2B market with the possibility to integrate various processes and gain insight into day-to-day business operations in real-time. They have developed modern Business Intelligence systems to analyze current data and historical facts to improve decision-making. The question remains, to what extent have software companies adopted the data-driven SPM approach by themselves and used rich data for decision-making in business, product and project management.

3 Research Methodology

The following research questions drive the first step in this study:

- **RQ1:** How is the decision-making process in software product management organized by software companies? To what extent are the SPM related decision-making processes and practices are formalized and evidence-driven?
- **RQ2:** What are the prime sources and types of pieces of evidence as well as models and tools used for decision-making in SPM?

- **RQ3:** What are the obstacles towards broader implementation of evidence-based software product management?

To address these questions, we made a study that can be classified as a positivist, exploratory multiple case study. The case sampling strategy was guided by the diverse case approach with its primary objective to achieve maximum variance along relevant dimensions [36]. Referring to the research questions, the goal is to identify decision-making practices and processes as well as to understand the logic behind them. To achieve that purpose, a within-case analysis was conducted with the analytical strategy of explanation-building based on the description of the cases, i.e., our study can be classified as exploratory case research.

We present all analysis in this paper in the form of propositions for further research. These propositions are grounded on qualitative data received through the series of semi-structured interviews with product managers and executives from five software companies. Companies selected for the study have their operation on EU and/or Russian markets. All companies have developed a SaaS solution for their customers. The SaaS solutions can be considered as mass-market services, where minor possibilities for customization are also available.

The data collection consisted of interviews that we consider as the first step of our longitudinal research project. The length of interviews varies from 2 to 3 hours. Their goal was to identify pressure points of decision-making in SPM, motivate companies to participate in the longitudinal study and assess both current status quo and product managers' perceptions of existing processes and practices. The data obtained covered the following topics:

- **General information about the company and products:** name, industry, market, number of employees, number of customers, maturity level, business model, number of products, products type and critical characteristics, product maturity level, etc.
- **SPM practices and processes:** SPM frameworks used, product activities allocation across business units, collaboration principles between business units, development approach, product manager's roles, and responsibilities, SPM tools used, SPM performance assessment principle, etc.
- **SPM decision-making principles:** formal regulation and written policies on SPM activities, factors of risks and uncertainty to consider by the product manager, types of data collected for SPM decision-making, models, and tools used to process provided data, information system support for SPM processes, etc.

4 Case Study

4.1 Companies Overview

A brief overview of the case companies is presented below and summarized in Table 1.

Company A provides a SaaS solution that includes a web service interface and mobile applications for different platforms. However, the company considers it as a single product. The product is highly dependent on the government's regulations. SPM duties are divided between two co-founders. One of them is responsible for business and product strategy, the vision of the firm and targeted market, while another co-founder is responsible for product functionality, product development and project management.

Company B offers highly specialized software fitting both the public and private market needs. The management believes that any possible market ceiling is far away, and the company will continue to grow at a rapid pace, driven by specific legislative initiatives taken in the EU that oblige other companies to use this type of software. The CEO is deeply involved in product management and responsible for defining product strategy, pricing policy, and other business-related issues. Product Owner is responsible for the rest of product management activities, mostly on tactical and operational levels.

Company C develops a fully integrated, software package for automating business processes such as accounting, ERP, CRM, etc. To date, the company has focused mainly on the local market, seeing more opportunities for organic growth within the country. Despite a large number of customers and the company's size, the company remains privately owned, which has a high influence on the culture of informal communication inside the company and decision-making through negotiations, including direct ones with company shareholders and top-level managers. The company has five product managers, each of whom is responsible for different modules in a single system. A product director working closely with the owners of the company is responsible for major strategic planning issues and defining the vision of the product.

Company D is a global player that offers accounting, CRM and industry-specific solutions for SMEs in Accountancy, Manufacturing, Professional Services, and Wholesale. Well-defined corporate governance practices and procedures are already established, and currently, the company is revising its software product management practices to make them more evidence-based. The company expects that it will improve decision-making and allow senior management to have better control over processes in the company and a better capability to assess product management performance.

Company E is a multinational company specializing in Internet-based services. The company provides clients with a variety of services, some of the provided services are technological platforms for taxi booking, work-at-home jobs search, food delivery, etc. Products form the unique ecosystem of services aimed to cover as many aspects of human life as possible and share the brand name, infrastructure and usually have are mutually integration. Still, the products vary widely regarding product management practices and development processes. The company holds oneself out as the analytical-driven; moreover, the analytical department is inter-product and, on request, provides analytical support to management processes.

Table 1. Characteristics of the five companies being analyzed

	CASE A	CASE B	CASE C	CASE D	CASE E
Ownership	Private	Private	Private	Public	Public
Number of Employees	<10	11 – 50	201 – 500	1 001 – 5 000	1 001 – 5 000
Number of Products (Modules)	1 (3)	1 (4)	1(8)	4 (> 25)	> 50
Market Type	B2B and B2C	B2B	B2B	B2B	B2B and B2C
Product Type	SaaS	SaaS	SaaS	SaaS	Platform
Number of Clients	> 100	> 1000	> 10 000	> 300 000	> 1 000 000
Organization structure	Lack of structure, confusion on roles and responsibilities	Roles and responsibilities are clear, still not formalized	Well established and formalized roles and responsibilities	Well established and formalized roles and responsibilities	Well established and formalized roles and responsibilities
SPM practices	Do not have a clear understanding of SPM, and do not use any particular framework	Have a clear understanding of SPM, but do not use any particular framework	Use externally-developed framework	Use externally-developed framework	Use the internally-developed framework
Interviewees	CEO, CTO	CEO, Product Owner	Product Director, Product Manager	Sr. Product Manager	Sr. Product Manager

4.2 Analysis

Consolidation of within-case analysis findings with a cross-case analysis facilitates a deeper understanding of the cases and accentuates the differences between them [37].

All companies are aware of evidence-based and data-driven management; still, all interviewees share the vision that this approach can be implemented fully only in large public companies with well-established organizational structure and available resources to establish the data-analytics business unit. Only **Companies D and E** were ready to implement a full-fledged evidence-based product management approach that includes formalization of evidence-based decision-making principles. Still, even for them it is a resource-intensive and non-trivial task: *"It took us more than a year to form a metrics system that we think is suitable to track the product development and measure it market performance... this work we did together with our analytical department and* *much has been done by analogy with existing metrics for other, more mature products"* (**Company E**).

At the same time, all interviewees specified that decisions in their companies are adequately grounded and to a certain extent data-driven. In **Company C** product managers try to support all sufficient decisions with analytics, all processes for obtaining pieces of evidence are not formalized. The company, despite the size, is trying to remain the spirit of the startup and afraid that formalization will reduce its ability to react to various market challenges and "*keep an ear to the ground*." The main reason for providing managers with broad responsibilities and opportunities in decision-making is part of the corporate culture and can be considered even as a competitive advantage that ensures solid growth: *"…the product manager can blow up the company. Definitely. Moreover, everyone here has full awareness of that… but we are growing perfectly. Until the situation remains, we do not need formalized and evidence-based processes"* (**Company C**).

The growth issue is even more important for SMEs. Both **Companies A and B** claimed that the introduction of evidence-based decision-making approach would possibly not only lead to a slowdown in their growth. Additionally, these companies specified that they have a lack of competences if dealing with the data available: "*We are collecting a lot of data, but simply not using it… everything falls on the shoulders of the members of our small team… we cannot afford hiring someone else, for now, to do this*" (**Company A**) and "*…numbers do not say anything, numbers just say that there is an issue, but they do not supply solutions or the way to go. You have to investigate and… If the information is not enough, emotions or intuition may help…"* (**Company B**).

Evidence-based software product management is considered primarily as an instrument of tactical SPM (or product planning according to ISPMA and similar frameworks [3]). **Companies B**, **C**, **D**, and **E** used various techniques for requirements prioritization, roadmapping and release planning that can be classified as evidence-driven. These processes require use of technical/structural data in decision-making, including feedback and bug reports collected by support unit, surveys with existing and potential clients collected by the sales team and key account managers, available log information regarding product usage.

The prime reason for product managers' perception on considering evidence-based SPM only on the tactical level seemed to be a lack of clear vision on what information could serve as proper evidence for strategic purposes. The majority of product managers considered evidence as a synonym to metric. However, more sophisticated pieces of evidence and data processing techniques are required for strategic decision-making. For instance, **Company C** confirmed that data related to sales and pricing is available, but not used for the decision-making: *"I have access to their CRM system...We have signals. If they are losing too many clients in the particular branch but... no, we do not work with this data...". Additionally,* **Company B** regrets that they are overworked and don't have competences to deal with such important source of evidence as knowledge collected through the analysis of the decisions made in the past: *"It could be interesting if we could take all the iterations for the last five years and you estimate and ask managers whether it was a mistake or not... I think that they are not able to do it right now...*

they are busier with the product...". Besides lack of vision towards sources of evidence and competences to work with them, even large companies feel that they can follow the leaders in strategic product management having better product and service quality as a strategic advantage: "The product is very successful, we have exponential annual growth… many decisions related to pricing and other economic-design issues were borrowed from the similar platforms" (**Company E**).

4.3 Discussion and Further Research Actions

We can formulate several propositions from our analysis in the form of answers to the identified research questions. Further field research should test these propositions in practice.

- **RQ1:** How is the decision-making process in software product management organized by software companies? To what extent are the SPM related decision-making processes and practices formalized?
  - The practice of decision making varies widely in software companies depending on their size and maturity level. Moreover, inside the same company, practices may vary from product to product, depending on the maturity level of the product and product managers competencies.
  - The transition towards formalized evidence-based decision-making starts with tactical and operational decision-making but rarely comes to strategic decision-making level. Tactical evidence-based SPM allow gradual development of the product and getting new consumers without violating the value of the product to the existing ones. Such thoroughness is usually not required in strategic SPM.
  - Formalized evidence-based decision-making processes are very resource consuming, regarding time, money, and people. Therefore, a company starts implementing them only after a particular stage of product (and company) maturity when the product is visible on the market and accepted by customers.
- **RQ 2:** What are the prime sources and types of pieces of evidence as well as models and tools used for decision-making in SPM?
  - Quantitative technical data is usually used as evidence for tactical and operational decision-making. However, strategic decision-making requires dealing with various sources of evidence that may also be of non-quantitative nature and challenging to quantify. An important source of evidence is accumulated past experience in the form of knowledge. In companies with well-established product management practices and processes, a lot of decisions on all three levels for new products are made taking into account prior experience.
- **RQ 3:** What are the obstacles towards broader implementation of evidence-based software product management?
  - There is a pervasive need for easy-to-use approaches and frameworks to support evidence-based SPM migration. Lack of clear vision regarding typology of evidences that could be served for informed decision-making could be names as prime obstacles towards broader implementation of evidence-based software

product management. These issues are supplemented by immature communication and knowledge sharing practices, poor integration between various SPM tools and systems used and scarcity of competency in data analytics, simulation modeling, and knowledge management.

The cross-case analysis revealed that it is possible to identify a clear trend towards the formalization of practices and processes for software product management, along with the company's development and growth. However, to the best of our knowledge, no attempt has been made in the academic literature to describe the logic of transforming product management practices towards evidence-based ones, including the critical milestones on this path. Development of the Product Management Maturity model that specify various aspects of transition towards evidence-based SPM could be used as a valuable supporting instrument. This model could complement other existing ones for development and operations [38] and project management [39, 40].

The in-depth interviews in five companies cannot produce a generalizable nomothetic theory [41]. Instead, we consider this qualitative study as idiographic, as it throws a glance on decision-making related to software product strategy in specific cases. To enhance the validity of this case study, further research is needed to shed light on current decision-making practices in the industry.

Besides a more substantial systematic study on current practices, SPM will benefit from a comprehensive review that will allow providing a typology of evidences for decision-making as well as methods for the analysis. This could consist of a rigorous theoretical and practical analysis of the power and limitations of available evidence-based SPM practices, methods, and techniques as well as development of a software product management maturity model with the focus on decision-making practices. This model could guide software companies in their transition towards evidence-based software product management that in term can reduce the likelihood of poor decision-making that leads to poor business success [42].

5 Conclusion

This study observed the current state of SPM decision making, managers' perceptions towards them as well as and the needs of the case companies. The presented perspective on decision-making practices complements and extends the existing literature on status quo and challenges in software product management [14, 15]. Software Product Management is a relatively young practice, and despite the presence of some significant research explaining its aims and objectives, the question of practical significance is still debatable. Software product managers have a strategic, cross-functional role that requires visibility into every phase of the product life-cycle. Immaculate product data, combined with the transactions surrounding each product, should, in theory, provide the product managers the insights they need to ensure product profitability and identify areas for improvement. Although there has been much discussion in the software business community on roles and area of responsibilities of software product managers, relatively little attention has been paid to the decision-making processes, practices, and principles.

This case study reveals that companies tend to try to formalize the existing decision-making practices to make them more transparent and evidence-driven. The more difficult question is that having the intention to move from intuition-based decision-making to a data-driven one, managers are often faced with the lack of a clear vision or understanding on what could serve as evidence in SPM and what techniques are required to make informed decisions. This becomes especially evident when dealing with strategic aspects of SPM associated with the product strategy and analysis of the product in relation to its market.

Acknowledgments

The first author has been partially supported by the Academy of Finland Grant no 310827 (MobiSPSULUT)

References

1. Ebert, C.: Software Product Management. CrossTalk - The Journal of Defense Software Engineering. 15–19 (2009).
2. Bekkers, W., van de Weerd, I., Spruit, M., Brinkkemper, S.: A Framework for Process Improvement in Software Product Management. In: European Conference on Software Process Improvement (EuroSPI) Proceedings. pp. 1–12 (2010).
3. Kittlaus, H.-B., Fricker, S.A.: Software Product Management: The ISPMA-Compliant Study Guide and Handbook. Springer-Verlag (2017).
4. Pragmatic Marketing: Pragmatic Marketing Framework. (2018).
5. Steinhardt, G.: The Product Manager’s Toolkit, Blackblot. (2017).
6. Maglyas, A., Nikula, U., Smolander, K., Fricker, S.A.: Core software product management activities. Journal of Advances in Management Research. 14, 23–45 (2017).
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 (EMC) Proceedings. pp. 338–343 (2002).
8. Mendes, E., Rodriguez, P., Freitas, V., Baker, S., Atoui, M.A.: Towards improving decision making and estimating the value of decisions in value-based software engineering: the VALUE framework. Software Quality Journal. 26, 607–656 (2017).
9. Fabijan, A., Olsson, H.H., Bosch, J.: The lack of sharing of customer data in large software organizations: Challenges and implications. In: International Conference on Agile Software Development (XP) Proceedings. pp. 39–52 (2016).
10. Ruhe, G., Saliu, M.O.: The art and science of software release planning. IEEE Software. 22, 47–53 (2005).
11. Denne, M., Cleland-Huang, J.: The Incremental Funding Method - A Data Driven Approach to Software Development. IEEE Software. 21, 39–47 (2004).
12. Olsson, H.H., Bosch, J.: Towards continuous customer validation: A conceptual model for combining qualitative customer feedback with quantitative customer observation. In: International Conference on Software Business (ICSOB) Proceedings. pp. 154–166 (2015).

13. Ngo-The, A., Ruhe, G.: Decision support in requirements engineering. In: Aurum, A. and Wohlin, C. (eds.) Engineering and Managing Software Requirements. pp. 267–286 (2005).
14. Fabijan, A., Dmitriev, P., Olsson, H.H., Bosch, J.: The Evolution of Continuous Experimentation in Software Product Development: From Data to a Data-Driven Organization at Scale. In: International Conference on Software Engineering (ICSE) Proceedings. pp. 770–780 (2017).
15. Ebert, C., Brinkkemper, S.: Software product management - An industry evaluation. Journal of Systems and Software. 95, 10–18 (2014).
16. Lawley, B.: Optimal Product Process, 280 Group. (2012).
17. Turoff, M., Hiltz, S.R.: Computer Support for Group Versus Individual Decisions. IEEE Transactions on Communications. 30, 82–91 (1982).
18. Cynert, R.M., Simon, H.A., Trow, D.B.: Observation of a Business Decision. The Journal of Business. 29, 237–248 (1956).
19. Benaroch, M.: Managing information technology investment risk: A real options perspective. Journal of Management Information Systems. 19, 43–84 (2002).
20. Ayers, D.J.., Gordon, G.L., Schoenbachler, D.D.: Integration and new product development success: the role of formal and informal controls. Journal of Applied Business Research. 17, 133–148 (2001).
21. Giardino, C., Unterkalmsteiner, M., Paternoster, N., Gorschek, T., Abrahamsson, P.: What do we know about software development in startups? IEEE Software. 31, 28–32 (2014).
22. Resceanu, I.C., Resceanu, C.F., Simionescu, S.M.: SaaS solutions for small-medium businesses: Developer's perspective on creating new SaaS products. In: International Conference on System Theory, Control and Computing (ICSTCC) Proceedings. pp. 140–144 (2014).
23. Hartmann, P.M., Zaki, M., Feldmann, N., Neely, A.: Capturing Value from Big Data - A Taxonomy of Data-driven Business Models Used by Start-up Firms. International Journal of Operations & Production Management. 36, 1382–1406 (2016).
24. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Currency (2011).
25. Blank, S.: The Four Steps to the Epiphany – successful strategies for products that win. K&S Ranch (2007).
26. 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 (LESS) Proceedings. pp. 1–15 (2013).
27. Briner, R.B., Denyer, D., Rousseau, D.M.: Evidence-Based Management: Concept Cleanup Time? Academy of Management Perspectives. 23, 19–32 (2009).
28. Rousseau, D.M.: Is There Such a Thing as "Evidence-Based Management"? Academy of Management Review. 31, 256–269 (2006).
29. McAfee, A., Brynjolfsson, E.: Big data: The Management Revolution. Harvard business review. 1–9 (2012).
30. Sheth, A.: Transforming Big Data into Smart Data: Deriving value via harnessing Volume, Variety, and Velocity using semantic techniques and technologies. In: IEEE

International Conference on Data Engineering (ICDE) Proceedings (2014).
31. Brynjolfsson, E., McElheran, K.: Data in Action: Data-Driven Decision Making in U.S. Manufacturing. (2016).
32. Brynjolfsson, E., McElheran, K.: The rapid adoption of data-driven decision-making. American Economic Review. 106, 133–139 (2016).
33. Girard, J., Girard, J.: Defining knowledge management: Toward an applied compendium. Online Journal of Applied Knowledge Management. 3, 1–20 (2015).
34. Fricker, S.: Software Product Management. In: Maedche, A., Botzenhardt, A., and Neer, L. (eds.) Software for People: Fundamentals, trends and best practices. pp. 53–81. Springer, Berlin, Heidelberg (2012).
35. Galli, B.J.: The Future of Economic Decision Making in Project Management. IEEE Transactions on Engineering Management. 1–18 (2018).
36. Seawright, J., Gerring, J.: Case Selection Techniques in Case Study Research. Political Research Quarterly. 61, 294–308 (2008).
37. Paré, G.: Investigating Information Systems with Positivist Case Study Research. Communications of the Association for Information Systems. 13, 233–264 (2004).
38. de Feijter, R., Overbeek, S., van Vliet, R., Jagroep, E., Brinkkemper, S.: DevOps competences and maturity for software producing organizations. In: International Conference on Evaluation and Modeling Methods for Systems Analysis and Development (EMMSAD) Proceedings. pp. 244–259 (2018).
39. Kwak, Y.H., Ibbs, C.W.: Assessing Project Management Maturity. Project Management Journal. 31 (1), 32–43 (2000).
40. Judgev, K., Thomas, J.: Project Management Maturity Models: The Silver Bullets of Competitive Advantage. Project Management Journal. 33, 4–14 (2002).
41. Cone, J.D.: Idiographic, nomothetic, and related perspectives in behavioral assessment. In: Conceptual foundations of behavioral assessment. pp. 111–128. Guilford, New York (1986).
42. Tomy, S., Pardede, E.: From uncertainties to successful start ups: A data analytic approach to predict success in technological entrepreneurship. Sustainability. 10, 602 (2018).

It Takes Three to Tango: Requirement, Outcome/data, and AI Driven Development

Jan Bosch[1], Helena H. Olsson[2] and Ivica Crnkovic[1]

[1]Chalmers University of Technology, Department of Computer Science & Engineering, Göteborg, Sweden
[2]Malmö University, Department of Computer Science and Media Technology, Malmö, Sweden

Abstract. Today's software-intensive organizations are experiencing a paradigm-shift with regards to how to develop software systems. With the increasing availability and access to data and with artificial intelligence (AI) and technologies such as machine learning and deep learning emerging, the traditional requirement driven approach to software development is becoming complemented with other approaches. In addition to having development teams executing on requirements specified by product management, the development of software systems is progressing towards a data driven practice where teams receive an outcome to realize and where design decisions are taken based on continuous collection and analysis of data. On top of this, and due to artificial intelligence components being introduced to more and more software systems, learning algorithms, automatically generated models and data is replacing code and the development process is no longer only a manual effort but instead a combination of human and automated processes. In this paper, and based on multi-case study research in embedded systems and online companies, we see that companies use different approaches to software development but that they often take a requirement driven approach even if they would benefit from one of the other two. Also, we see that picking the wrong approach results in a number of problems such as e.g. inefficiency and waste of development efforts. To help address these problems, we develop a holistic development framework and we provide guidelines on how to improve effectiveness in development. The contribution of this paper is two-fold. First, we identify that there are three distinct approaches to software development; *(1)* Requirement driven development, *(2)* Outcome/data driven development and *(3)* AI driven development and we outline the typical problems that companies experience when using the wrong approach for the wrong purpose. Second, we provide a holistic framework with guidelines for when to use what approach to software development.

Keywords: Requirement driven development, outcome/data driven development, AI driven development, holistic development framework.

1 Introduction

Today's software-intensive business is in the midst of profound changes in relation to development of software systems. With rapid pace, and across industry domains, sophisticated technologies for data collection and analysis are implemented to provide developers with real-time input on how the systems they develop perform in the field. Also, this data helps developers understand what functionality is used by customers and it allows product managers to confirm whether feature prioritizations were accurate [1], [2], [3], [4]. With automated practices for data collection and analysis, queries can be processed frequently to provide software developers and managers with rapid feedback and as a result, continuous improvements can be made to the systems. This reflects an interesting shift in that traditional requirement driven development practices that have been the de fault approach for decades [5], are being complemented by data driven development practices where teams use data to continuously improve and optimize the system to a certain outcome [4], [6]. As reported in previous research, the challenges with data driven development are numerous [7], but we can already now see that companies that are adept at acquiring, processing and leveraging data become more profitable as decision-making and prioritization based on accurate data from the filed can have a profound impact on annual revenue [4], [8].

Fueled by the increasing availability and access to data, artificial intelligence (AI) and technologies such as e.g. machine learning and deep learning are rapidly adopted in a variety of domains [9]. Although these methods and techniques have been in use for decades, recent years show an increasing use of these in industry with companies such as e.g. Google, Apple and Facebook leading the way but with software-intensive companies in the financial, the medical and the manufacturing domain as fast adopters. For these companies, and for any company with massive amounts of data, deep learning techniques are becoming a necessity and artificial intelligence components are rapidly complementing the traditional software components in a software system.

However, despite the rapid growth of data and the emergence of complementary approaches to software development, most companies have a strong tradition in requirement driven development. In this approach, system requirements are specified in the early stages of development, and although more agile requirements engineering practices are increasingly applied [10], the approach is characterized by a waterfall style of development that works well for systems where requirements are well understood and where revenue is based on delivering a complete product rather than continuous updates of software.

In our research, we see that companies use different approaches to software development but that there are a number of problems associated with selecting the most suitable approach. First, companies with a strong tradition in requirement driven development often take this approach even if they would benefit from an alternative approach. Second, proponents of outcome/data and AI driven development approaches tend to neglect other approaches and instead argue for their approach being the only right one. Third, picking the wrong approach for the wrong purpose results in a number of problems such as e.g. inefficiency and waste of development efforts. In this paper, and based on multi-case study research in companies in the embedded systems and in the online domain, we develop a holistic development framework including three distinct development approaches and we provide guidelines for how to improve effectiveness in development by selecting the optimal one.

The contribution of this paper is two-fold. First, we identify that there are three distinct approaches to software development; *(1)* Requirement driven development, *(2)* Outcome/data driven development and *(3)* AI driven development and we outline the typical problems that companies experience when using the wrong approach for the wrong purpose. Second, we provide a framework with guidelines for when to use what approach to software development.

The paper is organized as follows. In section 2, we detail the background of our research. In section 3, we describe the research method and the case companies involved in our research. In section 4, we report on the software development approaches in the case companies and we summarize our empirical findings. In section 5, we first identify three distinct development approaches and outline the key problems companies experience when picking the wrong approach for the wrong purpose. Second, we provide guidelines for when to use what approach. In section 6, we conclude the paper.

2 Background

Although the saying that things keep getting faster might sound a little worn-out, the fact is that the software business of today is experiencing bigger, and more rapid, transformations than ever before. The driving force of this is the increasing digitalization of industry that is disrupting companies and society in large to an extent that we have only seen the early beginnings of. As defined by Gartner [11], digitalization is *…" the use of digital technologies to change a business model and provide new revenue and value-producing opportunities; it is the process of moving to a digital business".* To survive this rapid change, companies need new capabilities such as e.g. 'speed' in terms of continuous deployment of software functionality. This allows for continuous collection of customer and product data to use as the basis for determining customer value of new products and services. Moreover, companies need 'data' to allow for artificial intelligence technologies such as e.g. machine learning and deep learning solutions to decrease the time it takes to manually shift through vast amounts of data and to have systems run automatic experiments to help identify, improve and even predict customer value. Finally, access and transparency to data allows for 'empowerment' and autonomy of teams that is critical for any company in order to advance and accelerate team performance and impact [12].

Interestingly, the transformations we see as a result of digitalization have an enormous impact not only on the products and the services that companies produce but also on the ways in which these products and services are produced, i.e. the development approaches themselves. As a result of digitalization and connectivity of products, the traditional requirements engineering process that has been the primary approach for software development for decades is being complemented with other approaches in which continuous use of data, rather than specification of requirements, informs development teams and software systems. As well-known to most software businesses, requirements engineering includes the identification of requirements and the modeling of these in order to develop an agreed upon understanding of what a future software system will look like in order to provide value to the customer and there exist a wide range of techniques to help the development team ensure that the requirements are complete [5]. As recognized in previous research, the goal of the requirements engineering process is to identify what functionality to build before development starts in order to avoid, or at least reduce the risk, of costly rework [5]. The reasoning is that mistakes that are revealed in the later stages of the development process are more expensive to correct, and that this can be avoided by identifying a stable set of requirements before development resources are allocated and system design and implementation activities start. More recently, agile practices have been adopted to improve flexibility and adaptability of the traditional requirements engineering process and to help software-intensive companies cope with increasing complexity in their software development processes [10].

However, with systems being connected to the Internet and technologies that facilitate data collection and analysis, we see that companies are increasingly complementing their traditional development approaches with other approaches. As one of the most influential trends in software industry, continuous deployment of software is challenging traditional ways-of-working in that it by-passes the notion of early requirements specification. Continuous deployment is a software engineering practice in which incremental software updates and improvements are developed, tested and deployed to the production environment on a continuous basis and in an automated fashion [13]. In this way, customer preferences and needs can be continuously collected, analyzed and deployed and rather than the traditional view of a system being finalized when delivered to customers continuous deployment allows for systems to evolve and improve over time and with delivery to customers as the starting-point for this. In online companies, continuous deployment of software and customer data from A/B tests are the norm for evaluating ideas and understanding customer value and with companies such as e.g. Amazon, eBay, Facebook, Google and Microsoft running thousands of parallel experiments to evaluate and improve their sites at any point in time [4, 13].

The trends described above reflect an interesting shift from a situation where traditional requirements engineering practices inform development of new features, towards a situation in which customer and product data is continuously collected and where companies use this data to inform development during run-time [1], [2], [4]. Also, this leads to interesting opportunities in the field of artificial intelligence as companies today possess such large data sets that manual processing of these become impossible. Today, machine learning and deep learning technologies are emerging as common components in what used to be traditional software systems and the development, production and organizational challenges associated with this shift are far from trivial [9]. Regardless, the software industry is in the midst of a transformation and in order for companies to stay competitive they need to understand, adopt and maximize the benefits from a number of different development approaches. As the systems they develop become connected and will include data collection and processing capabilities, artificial intelligence components and with continuous deployment of functionality as the way to deliver to customers, the approaches they use to develop these systems will advance too. With this in mind, we see the need for guidance on how to complement traditional requirement driven approaches to software development with other approaches as long-term success is seldom achieved by only substituting the former with the later but instead complementing existing expertise with new technology and skills.

3 Research method

The research reported in this paper builds on multi-case study research [14] in software-intensive companies in two industrial domains. The first domain is the *embedded systems domain* and here we studied companies in the context of Software Center (for detailed information please visit https://www.software-center.se/). These companies are large product development companies in e.g. the telecom, the automotive, the security camera, the defense and the manufacturing domains. As a common characteristic, the embedded systems companies are experiencing a challenging transition from being traditional product development companies to delivering products with associated services, and also purely digital services, and where connectivity and data are essential components for innovation and new business models. All companies in the embedded systems domain have significant experience and expertise in relation to requirement driven development as this has been the primary development approach for decades. Typically, the products and systems they develop are highly complex as they involve both hardware and software. In addition, they often have strict rules and regulations to follow as many of their products and systems operate in safety critical environments where standards such as e.g. ISO 26262 defines design, implementation, integration, verification, validation, and release. However, with increasingly connected products and with digital services that generate vast amounts of data, the embedded systems companies are starting to explore other development approaches that help them maximize the benefits associated with data. Although in complex and restricted environments, there are a number of emerging business opportunities and streams of revenue associated with data driven and digitalized services where traditional requirements driven approaches do not capture the potential of rapid feedback cycles and continuous deployment of software.

Over the years, our research collaboration with the Software Center companies has been reported in a large number of publications, e.g. [1], [2], [4], [15], [16] where additional details and careful company descriptions can be found. As reported in these papers, the transition towards digital products and services results in a number of challenges. As one of the most interesting ones, we see that the embedded systems companies seek to complement their traditional and requirement driven development approaches with other approaches in order to reap the benefits of the data they collect. In this paper, and based on our previous research in the Software Center companies, we explore the transition they are in and how the different development approaches they use complement each other.

The second domain is the *online domain* and here we studied companies developing online games, online payment services, media streaming services, travel and accommodation services, online search services and tools for developing artificial neural networks and adaptive systems. These companies are pure Software-as-a-Service companies and with revenue based on license fees, transaction fees and the digital products and services they produce. They continuously add software functionality to their products and they collect and use data as a basis for product development and improvements. In similar to the embedded systems companies, the online companies have access to large amounts of data and they are exploring different development approaches in order to maximize the benefits of this data. In contrast to the embedded systems companies, the online companies have less of a legacy in terms of requirement driven development. Even if this approach exists also here, they typically use data as the basis for development with teams receiving a quantitative target to realize and are asked to experiment with different solutions to improve a certain metric. In addition, some companies [e.g. …] use artificial intelligence and deep learning technologies as part of development in order to automate tasks and improve speed in problem-solving.

Our research collaboration with the online companies has been reported in a number of publications, e.g. [1], [4], [15], [16] and as reported in these papers, we see that data driven development practices including A/B testing and controlled feature experiments are well-established practices where collection and analysis of data works as the basis for decision-making and feature prioritization. In this paper, and based on our previous research in online companies, we explore the different development approaches they use and how these complement each other. In particular, we recognize how development approaches involving artificial intelligence and technologies such as e.g. machine learning and deep learning are emerging as critical components in many of the software systems they produce.

In total, our research collaborations with the different companies in these two domains cover a time period of more than seven years (2011 – 2018). The collaboration with the embedded systems companies has been an on-going engagement since 2011, and in relation to a number of different topics such as e.g. agile transformation, development feedback cycles, data driven development and value modeling of software features. The specific work on data collection and analysis, and how data can help improve software development, was initiated in 2015 and is on-going. The collaboration with the online companies was initiated in 2015 and is on-going. In all companies, and throughout this period, we have run frequent meetings, interview sessions and workshops involving project managers, product managers, product owners, software developers, software and system architects, data scientists, data analysts and a number of agile team coaches and scrum masters. Meetings are typically scheduled for one hour, workshop sessions for two – three hours and interviews for one hour. The empirical data we build on consists of hundreds of pages of interview transcripts, as many meeting and workshop notes, notes from informal meetings, thousands of e-mails and frequent telephone conversations. Throughout our research, we adopted an interpretive approach to data analysis with the intention to identify recurring elements and concepts in the transcribed interview protocols [17].

In this paper, we build on our previous findings from the embedded systems and from the online companies when exploring the different development approaches they use. In particular, we are interested in exploring how the development approaches they have traditionally been using are being complemented with other approaches.

4 Case study findings

In this section, and based on our previous research in embedded systems and online companies, we summarize our empirical findings in relation to existing and emerging approaches to software development. With selected examples from the two domains, we present the current state as well as the transition that the case companies are experiencing with regards to how to develop software products and services. In Table 1, we provide a summary where we generalize the characteristics of the development approaches in the two domains. It should be noted that the summary does not reflect details and deviations but rather it captures the dominant characteristics of each domain.

4.1 Software development approaches: The embedded systems domain

The embedded systems companies are in the midst of a challenging transition where the products they develop are rapidly becoming digitalized and where connectivity is key for future innovation and revenue. In this fast-changing environment, the hardware dependencies make development complex as the feedback cycles for hardware are slow while the software cycles are rapid. In most of the companies, the traditional and waterfall approach to development is applied in large parts of the organization while agile practices and methods such as e.g. Scrum are well-established in other parts. It should be noted that many of these companies offer a broad product portfolio which implies that the competence and expertise cover the development and delivery of physical products based on hardware components as well as digital services based on software components. To manage such a disparate product portfolio, the embedded systems companies apply a wide range of development methods and they need to constantly adopt new skills and ways-of-working. Still, and as the most common development approach, requirement driven development characterize both the mind-set and the organizational set-up in these companies. As a common practice, teams receive a requirements specification from product management, and the task for the team is to deliver according to specification. Even if many companies apply agile practices today, they have a long-standing and strong culture where requirements dictate development and where decisions and prioritizations are made based on previous expertise and experience. Typically, qualitative approaches are used to learn about customers with interviews, prototypes and observations being common techniques for data collection. Also, and in line with this culture, requirements are agreed upon in the early stages of development and with sudden changes being a costly disruptor and viewed as something to avoid.

However, and co-existing with the requirement driven culture, the embedded systems companies have been collecting data from their products well before they became connected as many of them are today. For example, the automotive companies started collecting diagnostics data from vehicles already in the early 90’s to use as the basis for maintenance whenever a truck or a car was taken to a garage for service. More recently, and as a result of vehicles becoming connected to the Internet and with practices such as continuous deployment in place, car manufacturers can push software updates to the vehicle on a continuous basis without taking the vehicle out of traffic. This allows for preventive maintenance and has become key to prolong the lifetime of a vehicle and avoid costly repairs. Also, effective use of data allows car manufacturers to detect errors while the vehicle is running and before the customer is even aware of them. In similar, telecom companies collect huge amounts of traffic and configuration data as the basis for optimizing performance and operation of their systems as well as for predictive maintenance and monitoring. Based on our research, we see that many of the embedded systems companies are in the process of instrumenting their products to increase and further improve data collection and analysis practices. Also, there are examples of A/B testing initiatives where companies run experiments with customers to determine whether version A or B of a software feature is the optimal and most appreciated one [18]. In all the companies we studied, the collection and increasing use of data has started to affect the traditional role of product management. With an increasing flow of customer and product data, development teams get a new source from which they learn about the products they develop. In similar, product management get an opportunity to use this data for understanding what adds value to customers. In previous work [2], we report how traditional roles such as e.g. product management change as new roles such as e.g. data scientists emerge. In this research, we see that as companies advance in extracting value from the data they collect, this data will become an effective means for decision-making, as well as work as a basis for product improvements and innovations.

Based on our most recent interactions with the embedded systems companies, we see an emerging interest in artificial intelligence and associated technologies. With connected systems and with large data sets available, new opportunities arise in terms of how to manage, process and utilize this data. For many of the companies, automated practices for collection and analysis of data are already in place as continuous integration and deployment are becoming critical components of their software development approaches. Still, however, supporting infrastructures for increasingly big volumes of data that can handle complexity in terms of variety and velocity [9] are rare and something that would be needed for effective use of solutions such as e.g. deep learning. In our experience, and based on current practices in the case companies, real-time processing of data and artificial intelligence components for supporting this will have a significant impact on future business opportunities as well as for the way in which these companies develop software.

4.2 Software development approaches: The online domain

In contrast to the embedded systems companies, the online companies are less frequent users of requirement driven practices. Although they exist, they don't serve their purpose as the products and systems the online companies develop are inherently different in characteristics and therefore, require other development approaches. Instead, practices such as continuous integration and deployment are fully in place and with products being digital there are no hardware dependencies that slow down the development cycle. This reduces complexity and increases speed and in the majority of the companies, new software functionality is released on a daily or weekly basis. Instead of requirements, the online companies use data collected from their products as the basis for understanding customer needs and preferences. In our experience, most of the online companies have instrumented their products in order to collect relevant data and they have software tools that help them analyze this data. As the basis for data collection, they run A/B tests in which hypotheses on what adds customer value are validated. A/B tests are experiments where two versions of software functionality are compared to determine which one performs the better in relation to predefined criteria such as e.g. conversion rate, click rate or time to perform a certain task [4]. To collect relevant data, users' interaction with the system is instrumented and data on e.g. page views, clicks etc., is collected. In this way, the online companies monitor click-through rates, number of sessions per user, revenue per user and other metrics and use statistical analysis to determine which variant performs better for a given conversion target [8], [15], [19]. In some of the companies, hundreds of experiments are run in parallel at any point in time and a large number of metrics are used to track product performance and user behaviors. With this data available, the online companies have the opportunity to respond fast and base decisions and prioritizations on data rather than on previous experience and expert opinions in the company. Currently, A/B testing is the dominant technique for optimizing performance, validating new concepts and test new ideas.

Despite the many advantages with using data as the basis for development, the online companies experience challenges with this approach just like the embedded systems companies experience challenges with their requirements driven practices. As reported in our previous research [4], [15], [16] to scale the impact of experiments, to identify and agree on key metrics to optimize for and to find effective mechanisms for evaluating the success of an experiment are difficulties that the online companies face. Also, and as the volume of the data sets increases, there is the need to advance the storage and processing capabilities as well as adopt mechanisms that manage variety and velocity of data.

In our most recent research, we have had the opportunity to learn about some of the emerging trends in these companies and especially about their rapid adoption of artificial intelligence and deep learning solutions. These technologies enable radical improvements in the development cycle by increasing the effectiveness of development and by reducing development time of novel functionality. In one of the case companies [9], deep learning components are developed to provide companies in a variety of domains (e.g. real estate, bookkeeping, weather forecasting etc.) with a platform and with tools for processing, modelling and recognize and predict patterns in large data sets. However, and as recognized in [9], [20], there are no systematic and repeatable methods for creating, evolving and maintaining software systems using these technologies and although successful instantiations exist there are a number of challenges to solve before online companies, as well as embedded systems companies, can fully benefit from artificial intelligence as part of their daily development practices.

Table 1. Generalization of characteristics of the current software development approaches in the two domains.

Characteristic	Embedded Systems Domain	Online Domain
Development cycle	Long (project based)	Short (sprint based)
Requirements cycle	Project/sprint	Sprint/continuous
Quality assurance cycle	Discontinuous	Continuous
Release frequency	Monthly/yearly	Daily/weekly
Decision-making	Expert driven	Data driven
Value creation	Infrequent (product/system)	Frequent (feature/functionality)
Value assessment	Internal validation	External validation

5 Towards a holistic development framework

In this paper, we explore the transition that companies in the embedded systems and in the online domain are experiencing due to digitalization of products and services. Based on our previous research in a large number of companies, we see that companies are complementing their traditional development approaches with other approaches and that this requires a careful understanding of when to use what approach. Below, we identify three distinct development approaches that we see exist in the companies we studied. We summarize the approaches in Table 2. Furthermore, and as an inductively derived model from generalizing our case study findings, we provide a holistic development framework (Figure 1) with guidelines for when to use what approach to software development.

5.1 Three software development approaches

Based on our case study research, we identify that companies use three different approaches to software development. First, they use a *requirement driven development* approach where software is built to specification and where product management is responsible for collecting and specifying requirements as input for the development teams. As can be seen in the empirical examples, this development approach is predominantly used when the new features or new functionality are well understood and defined and where business revenue is not based on frequent releases of new functionality. Especially in the embedded systems companies, there is a long and well-established practice of developing software systems based on requirements. In all these companies, requirements are collected, specified and carefully documented as the main input for development teams and as the mechanism to confirm that a system is developed and delivered according to customer preferences and needs [5], [10]. Over the years, and as experienced in our case companies, a number of limitations have been recognized in relation to the requirement driven development approach with the assumption that customer requirements can be identified before development starts as the most questioned one. Also, techniques and tools for eliciting customer requirements are often insufficient as these tend to focus on what customers say they want rather than what they do in practice which causes a situation in which requirements that can be made explicit are the only ones that can be captured while the more implicit ones remain invisible. Finally, the companies we studied confirm that with techniques such as e.g. brainstorming, interviews, focus groups, observations and prototyping, the amount of data that is collected is relatively small and primarily qualitative in nature which makes it difficult to generalize and identify patterns of behaviors of a large customer group.

However, the requirement driven development approach is well suited for situations in which features and functionality are well-understood and where there is a long-term agreement between the customer and the development organization. Typically, this approach applies for products and services that are intended to last over time and where there is less frequent change imposed on the system. When applied in fast changing environments where customer requirements fluctuate, the requirement driven development approach should be avoided as it falls short on managing frequent iterations and short development cycles.

The second approach companies use is the *outcome/data driven development* approach where development teams receive a quantitative target, i.e. an outcome, to realize and are asked to experiment with different solutions to improve the metric. This development approach is predominantly used for development are new features that are used frequently by customers, and for innovation efforts when there is uncertainty on how to realize a new feature. As can be seen in the empirical examples, this approach is the dominant approach in the online companies where development teams are assigned a certain metric, e.g. conversion rate, and are responsible for improving this metric. To do so, a team typically runs A/B tests with selected customers to identify what version of a website that improves the metric and that moves the needle in the direction set by the business. The decision is based on data that is collected during the experiment and the approach differs from the requirement driven development approach in that continuous collection and analysis of customer and product data informs development rather than requirements specified in the early stages of development. Also, while the requirement driven approach is characterized by smaller amounts of qualitative data as the basis for decision-making, the outcome/data driven approach uses large and quantitative data sets collected at run-time and by instrumenting the code to monitor specific metrics. In our research, we see example of this approach also in the embedded domain where experimentation with different software solutions are becoming increasingly important to determine and validate customer value [1], [2], [21]. The companies we studied report on a number of challenges involved in outcome/data driven development. Often, these relate to the challenge with accumulating and scaling the impact of experiments [4]. In the companies we studied, experiments support smaller improvements of features rather than having an impact on high-level business decisions such as larger re-designs, new product development or innovation initiatives and impact of an experiment is limited. In combination with poor evaluation criteria, the trustworthiness of the experiment might be low. Still, the outcome/data driven development approach is well suited for situations where there is a need to test different hypotheses and where the solution to a problem is unclear. Also, the approach is often applied in innovation efforts as there is the need to test and trial with customers in order to identify the potential value of a new feature, a new product or a new service.

The third approach companies use, and that is rapidly emerging as a new approach to software development, is the *AI driven development* approach where the company has a large data set available and uses artificial intelligence techniques such as machine learning and deep learning to create components that act based on input data and that learn from previous actions. In the case companies we studied, AI is perceived as a very powerful approach with the potential to take on far more complex assignments humans by augmenting our skills, talents and abilities. Examples of this type of development include e.g. object recognition is autonomous cars as well as speech recognition in modern user interfaces. As another example, one of the case companies refers to the use of AI for predicting business sales by pulling data from sales tools together and by using patterns found in this historical and rich data set. Typically, this approach applies for product and service development in which a company has access to a large data set with very many data points and where minimizing prediction errors is critical. Also, and as recognized in the case companies, it is an approach well suited for development situations in which there are too many potential alternatives that manual processing of these would be either too difficult, too time consuming or too expensive. As the AI approach is fundamentally different from traditional software development in that much of the responsibility for finding a solution to a problem is left to the computer, problems arise when an organization e.g. lack mechanisms and infrastructures for running experiments, has limited resources for large and complex data sets and when the organizational culture, skills and interests do not align with the cross-functional collaboration that is critical for building a production-ready AI system. As recognized in [9], there are additional challenges related to development, production and organization and as this development approach is still in its infancy in many of the companies we studied we foresee significant work in the area of defining systematic and repeatable methods for creating, evolving and maintaining systems using AI techniques.

Table 2. Summary of the current software development approaches that are used in the two domains.

Development approach	Definition
Requirement driven development	Software is built to specification. This development approach is predominantly used when new features or functionality are well understood and defined.
Outcome/data driven development	Development teams receive a quantitative target to realize and are asked to experiment with different solutions to improve the metric. Examples of this development approach are new features (used frequently by customers) and innovation efforts.
AI driven development	A company has a large data set available and use artificial intelligence techniques such as machine learning and deep learning to create components that act based on input data and that learn from previous actions. Examples of this development approach include e.g. object recognition in autonomous cars and speech recognition in modern user interfaces.

5.2 Holistic development framework

As reported above, and due to the increasing access and availability to data, companies are starting to complement their requirement driven development approaches with other approaches. With the adoption of agile development practices [22], [23] the companies we studied have been able to shorten their internal development cycles and, in many cases, integrate development with product operation. In these companies, it is possible to iteratively build new functionality and continuously measure to what extent this functionality is delivering on the expected outcomes. In addition, recent developments in artificial intelligence allow for radical improvements in the development cycle in terms of effectiveness and exploration of novel functionality.

However, and as recognized in this research, the integration of these development approaches is not well understood and there exist little guidance for when to select one approach over another. In addition, and as future systems will include both traditional software components as well as artificial intelligence components, the combination of approaches will be critical as most software organization will have to manage not only one approach but all three.

To help address this challenge, and to provide companies with a framework on when to select what approach, we present a holistic development framework (Figure 1). Based on a generalization of our case study findings, we outline the three development approaches and we identify the purposes for which each approach is optimal. As can be seen in this framework, the (1) requirement driven development approach is well suited for regulatory features, for competitor parity features and for commodity features, the (2) outcome/data driven development approach is well suited for value hypotheses, development of new "flow" features, i.e. features used frequently by customers and for innovation and the (3) AI driven development approach is well suited when aiming to minimize prediction errors, when there are many data points and when there is a combinatorial explosion of alternatives.

The framework pictures an overall development environment where the system in operation consists of traditional software components as well as AI components, and where continuous deployment practices allow for behavior data to be continuously collected and used as the basis for development. In this environment, and with systems involving different components, the key challenge is to effectively select, combine and deploy different development approaches. Although successful instantiations exist in research and industry, there are no systematic, repeatable methods for creating, evolving and maintaining systems using these techniques.

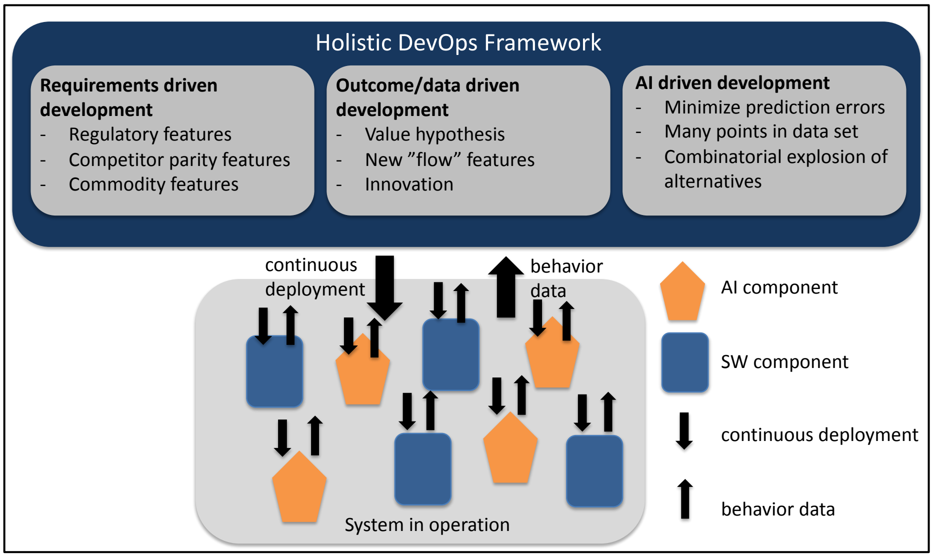


Figure 1. The 'Holistic DevOps Framework' including the three approaches to software development.

6 Conclusions

Today's software industry is in the midst of dramatic transformations with digitalization challenging existing ways-of-working. With increasingly connected and intelligent products, and with availability and access to massive amounts of data, the traditional requirement driven approach to software development is being complemented with other approaches that reflect these new opportunities and technologies. However, in our research we see that companies often take a requirement driven approach even if they would benefit from one of the other two. Also, we see that picking the wrong approach results in a number of problems such as e.g. inefficiency and waste of development efforts.

In this paper, and based on multi-case study research in embedded systems and online companies, we identify three distinct approaches to software development: *(1)* Requirement driven development, *(2)* Outcome/data driven development and *(3)* AI driven development and we provide a framework with guidelines for when to use what approach to help minimize the problems associated with using the wrong approach for the wrong purpose. With this framework, we aim to help companies effectively select, combine and deploy different development approaches in order to manage the digital transformation they are in.

In our future work, we intend to further validate this framework and explore how software-intensive companies in different domains can benefit from complementary development approaches and how successful selection of approaches can become key for competitive advantage.

References

1. Olsson, H.H., and Bosch, J. (2017). Towards Evidence-Based Development: Learnings from Embedded Systems, Online Games and Internet of Things. *IEEE Software,* October 2017, Issue 99.
2. Olsson, H.H., and Bosch, J. From Opinions to Data-Driven Software R&D: A Multi-Case Study on How to Close The 'Open Loop' Problem. *In Proceedings of EUROMICRO, Software Engineering and Advanced Applications (SEAA), August 27-29, Verona, Italy, 2014.*
3. Bosch, J. 2012. Building Products as Innovations Experiment Systems. *In Proceedings of 3rd International Conference on Software Business, June 18-20, Cambridge, Massachusetts.*
4. Fabijan, A., Dmitriev, P., Olsson, H. H., and Bosch J. (2017). The Evolution of Continuous Experimentation in Software Product Development: From Data to a Data-driven Organization at Scale. *In Proceedings of the 39th International Conference on Software Engineering (ICSE), May 20 – 28th, Buenos Aires, Argentina.*
5. Pohl, K. (2010). *Requirements engineering: fundamentals, principles, and techniques.* Springer Publishing Company, Incorporated.
6. Bosch, J. (2016) Future Trends in Software Engineering. *IEEE Software.*
7. Dmitriev, P., Frasca, B., Gupta, S., Kohavi, R., and Vaz, G. (2016). Pitfalls of Long-Term Online Controlled Experiments. *In Proceedings of IEEE Conference on Big Data, December 5 – 8, Washington.*
8. Kohavi, R., and Longbotham, R. (2015). Online Controlled Experiments and A/B Tests, *In Encyclopedia of Machine Learning and Data Mining, pp. 1–11.*
9. Arpteg, A, Brinne, B, Crncovic-Friis, L, Bosch (2018). Software Engineering Challenges of Deep Learning, *In Proceedings of the 44th Software Engineering and Advanced Applications Conference (SEAA),* August 29 – 31, Prague, Czech Republic.
10. Paetch, F., Eberlein, A., and Maurer, F. (2003). Requirements Engineering and Agile Software Development. *In Proceedings of the 12th IEEE International Workshop on Enabling Technologies, p. 307-313.*
11. https://www.gartner.com/it-glossary/digitalization (Accessed October 5th, 2018).
12. Bosch, J. (2016). *Speed, Data, and Ecosystems: Excelling in a Software-Driven World.* Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series. CRC Press.
13. Savor, T., Douglas, M., Gentili, M., Williams, L., Beck, K., and Stumm, M. (2016). Continuous deployment at Facebook and OANDA. *In Proceedings of the 38th International Conference on Software Engineering Companion, pp. 21-30.*
14. Maxwell J. A. 2005. *Qualitative Research Design: An interactive approach*, 2nd Ed. Thousands Oaks, CA: SAGE Publications.
15. Fabijan, A., Dmitriev, P., Olsson, H.H and Bosch, J (2018). Effective Online Experiment Analysis at Large Scale. In Proceedings of the 44rd Euromicro Conference on Software Engineering and Advanced Application, SEAA'18, Prague, Czech Republic, 2018.
16. Mattos, I. D, Dmitriev, P., Fabijan, A., Bosch, J and Olsson, H.H (2018). Beyond Ad-Hoc Experiments: A Framework for the Online Controlled Experimentation Process. *In Proceedings of the 19th International Conference on Product-Focused Software Process Improvement, (PROFES). November 28 – 30, Wolfsburg, Germany.*
17. Walsham, G. (1995). Interpretive case studies in IS research: Nature and method, European Journal of Information Systems, vol. 4, pp. 74-81.
18. Bosch, J., and Eklund, U. (2012). Eternal embedded software: Towards innovation experiment systems, In Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change, p. 19-31, Springer; Berlin, Heidelberg.

19. M. Kim, T. Zimmermann, R. DeLine, and A. Begel, (2015). The Emerging Role of Data Scientists on Software Development Teams, No. MSR-TR-2015–30, pp. 1–10
20. Submitted paper…
21. Olsson H. H., and Bosch J. 2013. Towards Data-Driven Product Development: A Multiple Case Study on Post-Deployment Data Usage in Software-Intensive Embedded Systems. *In Proceedings of the Lean Enterprise Software and Systems Conference (LESS), December 1-4, 2013, Galway, Ireland.*
22. Highsmith, J., and Cockburn, A. (2001). Agile Software Development: The business of innovation, Software Management, September, pp. 120-122.
23. Larman, C. (2004). Agile and Iterative Development: A Manager's Guide. Addison-Wesley.

Do software startups innovate in the same way? A case survey study

Jorge Melegati and Xiaofeng Wang

Free University of Bozen-Bolzano, Piazza Domenicani 3, Bolzano, Italy

Abstract. The research interest in software startups has expanded a lot in the last years as shown by the increase in the published papers, and the organization of research workshops. However, two recent systematic mapping studies recognize an inconsistency in the characterization of software startups in the literature, even though they acknowledge that innovativeness and uncertainty are the common themes the literature uses to describe these companies. In the new product development literature, even though not consolidated, innovativeness is usually related to technology and/or market discontinuities. These two different types of novelty could bring distinct consequences to software development activities in software startups. Using a case survey research approach, we analyzed 27 published papers from the period 2013-17. We identified and categorized the innovation in 18 software startups products or services from the perspective of discontinuities. We found that software engineering literature did not differentiate software startups based on the innovations they develop. Nevertheless, most studied software startups work on products with a market discontinuity and without a technological one.

Keywords: software startups · innovativeness · technology discontinuities

1 Introduction

Research in software startups has grown in the last five years as discussed by Berg et al. in a recent systematic mapping study [2]. Comparing the papers in this period to a previous systematic mapping study performed by Paternoster et al. [14], the authors conclude that the rigor of primary studies has increased in this period. Nevertheless, they also found an inconsistency of characterizing software startups similarly to what Paternoster et al. [14] have found. In this more recent study, the authors performed a thematic analysis of the software startup term used in research and their results indicated that no single factor was used by all papers to characterize software startups. They concluded that the lack of a proper definition make it challenging to develop a body of knowledge for software startup context. Similarly, Unterkalmsteiner et al. [17] proposed a research agenda for software startups and, in one of the tracks, remembered that software startups definitions are "not granular enough [...] making the transfer of practices from company to company difficult."

Still, in Berg et al.'s thematic analysis, the terms most used were "innovation/innovative" and "uncertainty" [2]. This relation between innovation and uncertainty is explored in new product development literature. Salomo et al. [15] mentioned "high innovative development may evolve along an unexpected path, thus requiring frequent or continuous information updating and generation of new information". Nevertheless, in innovation literature, product innovativeness has several conceptual configurations [11]. Garcia and Catalone [7] performed a literature review on innovation and product innovativeness terminologies and found several categorizations to label degrees of innovativeness. Nevertheless, according to the authors, "a single theme, however, does underlie all these classifications of innovations: innovativeness is a measure of discontinuity in the status quo in marketing factors and/or technology factors". Although Calantone et al. [4] concluded that product innovativeness has no direct effect on new product profitability, different degrees of newness and discontinuities change factors in new product development (NPD) processes [7].

To the extent of our knowledge, there is no discussion in software startups literature about the degree of newness of products developed by studied companies. Then, this study investigates the following research question:

RQ: What defines the innovation of a software startup?

To answer the research question, we performed a case survey on the papers published in the period 2013-17 covered by Berg et al. [2] in their recent systematic mapping study. We read all the papers, selected studied software startups that got the information that allowed us to analyze product and business models, and categorized them accordingly. The results indicate that scholars have studied companies developing products with discontinuities from technical and market perspectives without distinction. This lack of homogeneity could hinder generalization of results presented.

The remaining of this paper is organized as follow. Section 2 presents related work on the software startup term definition in software engineering literature and presents technological innovativeness as discontinuities, Section 3 describes the methodology used to perform this study and Section 4 presents the results and discuss them. Finally, Section 5 concludes the paper and proposes future work.

2 Background and Related work

The scientific interest on software startups has grown in the last years but the definition of what is a software startups is still not consolidated. Section 2.1 displays how this problem has been discussed in secondary studies on the topic. Given that, technological innovativeness is one of the most used themes to describe these companies, Section 2.2 presents some definitions from new product development (NPD) literature.

2.1 Startup definition

Sutton [16] is responsible for an early characterization of a software startup company back in 2000. According to him, these companies are widely represented by youth and immaturity, limited resources, multiple influences and dynamic technologies and markets. In their 2014 systematic mapping study, Paternoster et al. [14] concluded that there was no agreement on a standard definition and identified the most frequent themes used to characterize software startups: lack of resources, high reactiveness and flexibility, innovation, uncertain conditions, time pressure and fast growth. In the more recent SMS, Berg et al. [2] performed the same analysis and reached a similar result. Nevertheless, the most used themes were now innovation/innovative, uncertainty, small team and lack of resources.

2.2 Technological innovativeness

In a seminal literature review about innovativeness, Garcia and Calantone [7] recognize that the term was still not homogeneous in the new product literature, including what is considered what is new. Nevertheless, the authors recognize a consistency: innovativeness “is always modeled as a degree of discontinuity in marketing and/or technological factors.”

The authors also emphasize that innovativeness should be analyzed from two different perspectives: a macro, related to the newness of that product to the outside of the firm, and a micro, related to the novelty to the firm. Based on these perspectives, they define:

- radical innovations as representing a technological and marketing factors in a macro perspective;
- really new innovations as showing a technological *or* a market discontinuity; and
- incremental innovations as those presenting any discontinuity only in a micro perspective.

Given a startup is generally considered a new company with little or no operating history, we can limit our analysis to the macro perspective, that is, only radical or really new innovations.

3 Research method

As Larsson [9] mentioned in his paper about the methodology, the case survey is an inexpensive and powerful method to identify and statistically test patterns across studies. The author describes the basic procedure to perform such study:

- to select a group of existing case studies;
- develop a coding scheme to systematically analyze and convert qualitative descriptions into quantified variables;
- based on the coding scheme, several raters code the cases;

– analyze the quantitative data.

Then, the first step was to select the primary studies to be analyzed. Given that Berg et al. [2] published a systematic mapping study on software startup engineering on *The Journal of Systems & Software*, the primary studies they identified are high probably the most important also to this study. Another interesting feature of this study is that it separated the papers from period 2013-2017 against the whole studied period from 1994-2013. Since this study is focused on how current researchers are selecting software startups to study and that the authors also concluded that the rigor has increased in this period in comparison to before, it is reasonable to focus on this more recent time period.

Then, we got the 27 full text papers and read them carefully. Only 7 of those provided information that allowed us to describe the business model of, at least, some studied software startups. Table 1 displays the papers that contained at least one describable software startup and the number of software startups described in each of them. Based on the descriptions and the authors, we could infer that some software startups were mentioned more than in one paper. Then, their descriptions were merged and analyzed as one. The total number of descriptions was 23 but only of those 18 were unique cases.

Table 1. List of empirical papers on software startups reviewed.

Paper	Number of software startups described
Giardino et al. [8]	2
Nguyen-Duc et al. [12]	3
Bajwa et al. [1]	4
Nguyen-Duc et al. [13]	5
Marks et al. [10]	1
Chanin et al. [5]	2
Duc et al. [6]	6

The software startups' descriptions were extracted and classified according to if they present or not technological and market discontinuities. According to Garcia and Calantone [7], a technology discontinuity represents "a paradigm shift in the state of science or technology in a product" and a market one "may require new marketplaces to evolve, and/or new marketing skills". In our classification, a new product or service to contain a technology discontinuity, it had to represent a product that demanded the creation a new technology like a prediction tool for financial markets. Meanwhile, a market discontinuity represents the application of a well-known technology (like web development) into the creation of a new product, or the application of an existent product to a new market. Products could also display both types of discontinuity when its creation demands a new technology to be applied in a product without a similar solution in the market like a new real-time solution to support sales.

4 Results and discussion

Table 2 displays a summary of the software startups descriptions presented in the analyzed papers and if they present or not a technical or market discontinuity. Figure 1 depicts a graphical representation of the analysis. In the lower-left quadrant where a product would not have neither a technical nor a market discontinuity is empty. This was expected: software startups as innovative enterprises should present at least one discontinuity. Most of startups (13 out 18) are in the lower-right quadrant, that is, they display a market discontinuity without a technical one. The companies apply well-known technologies to create a product to tackle a problem in an existent market or they pretend to create a new market. Meanwhile, only 2 of them are in the upper-left quadrant, showing a technical discontinuity without a market one, that is, they are focused on the creation of a new technology to tackle an existent problem, a new solution to compete with existent products or services. The remaining 3 software startups are in the upper-right quadrant presenting both types of discontinuities: they develop a new technology to tackle a problem that has not been solved before. This last group represents the most challenging environment to a software startup.

In relation to software development challenges, the two groups that present only one type of discontinuity could be also summarized as follows:

- Only technical discontinuity: these teams develop brand-new technologies for some consolidated markets. It was possible to observe that their members generally have strong research backgroun and some of these companies were created in universities. Their challenges are situated more in technical problems than customer/market problems. .
- Only market discontinuity: these teams develop new products or services with well-know technologies like websites or mobile apps. We observed that these companies are generally formed by recent graduated students. Their challenges are situated more in market problems that is, their major risks are finding customers or users.

These two clusters could also be viewed as teams trying to solve different issues on the problem-solution space. The first group focus on a *solution* issue to a well-defined or well-known problem meanwhile the second group focus on finding the right *problem* to tackle. For this group, once the problem is found, the solution is, at least at a higher level, straightforward. Those software startups which product presents both discontinuities face problems from both spaces.

This difference influences the software development process in these companies. For instance, if the problem is not well-understood, requirements engineering practices should be less effective and the team may need to use other techniques like, for instance, brainstorming and ideation. That is, they can take more advantage of Customer Development and other lean startup practices [3].

Table 2. List of software startups described in the literature.

Number	Software Startup	Description	Discontinuity	
			Technical	Market
1	Milkplease [8]	Deliver grocery shoppings from local supermarkets to customers' doors with the collaboration of neighbors		X
2	Picteye [8]	On-line service to sell pictures of public and private events		X
3	Story A [12], StartupC [13]	A platform to sale photos to local event participants		X
4	Story B [12]	A sonar system to produce video-type imaging underwater where normal cameras don't work	X	X
5	Story C [12], Hooka [1]	An online ticketing system focused on small companies that cannot afford expensive solutions		X
6	Dicy [1]	Video service for other startups to create their own promotional videos		X
7	DocMine [1]	Unified API to access different social media sources		X
8	EasyLearning [1], Startup A [13], CT5 [6]	Game-based learning platform for teachers give quiz to students in a classroom		X
9	StartupB [13]	Real-time sale support solution	X	X
10	StartupD [13]	App for share meal		X
11	StartupE [13], CT2 [6]	A mobile solution that allows different departments of a construction project to collaborate.		X
12	Optimality Technologies [10]	Software tool to database modernization	X	
13	StartupA [5]	Mobile application to link organic food producers to consumers		X
14	StartupC [5]	Online invest platform with predictive capabilities	X	
15	CT1 [6]	An spin-off of a social media corporation that develops a hyper-local news platform.		X
16	CT3 [6]	To facilitate events organization and tickets purchase in Norway.		X
17	CT4 [6]	An Airbnb style solution (sharing economy) for shipping services.		X
18	CT6 [6]	An IoT solution to be used by fish farms for tracking and management.	X	X

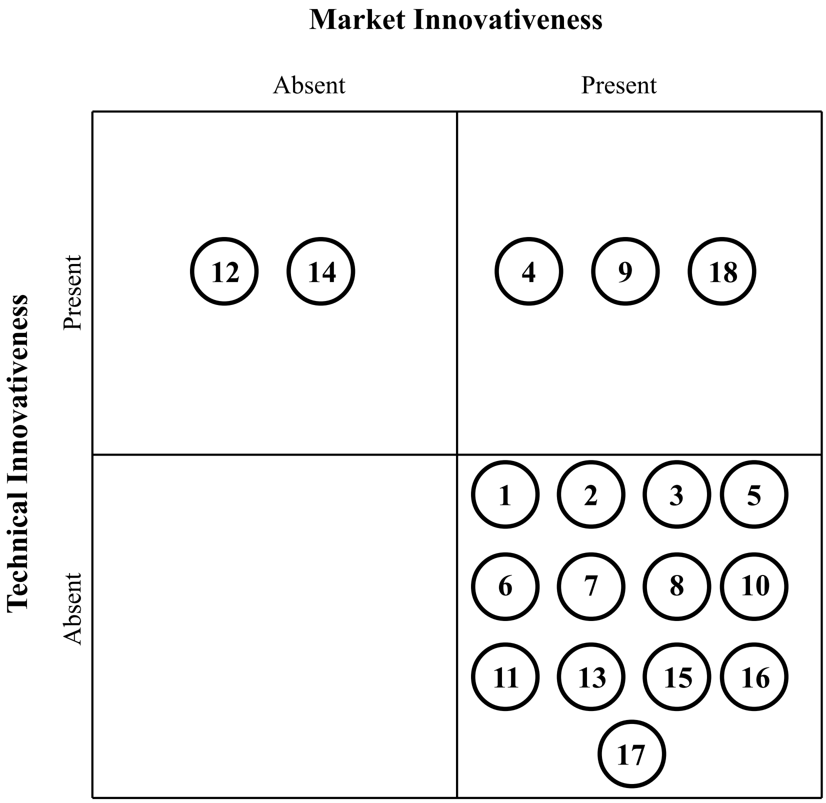


Fig. 1. Diagram of software startups innovativeness

Whereas if the customer problem is clear and the solution is not well-understood and need to be developed, the focus will be more in software design and the implementation itself.

4.1 Limitations

This paper has as a clear limitation based on the reduced information provided in the papers on how software startups studied create value to their users or customers. Although this lack of details could make difficult to classify the products, even this limited amount of information was enough to determine the discontinuities present. It is possible, tough, that we were mislead and some software startups were wrongly classified. Nevertheless, it is highly unlikely that several were wrong at the point to change the main result: software startups that face different discontinuities are grouped and have their practices analyzed together.

To increase validity and reliability, both authors did the classification and compared their results. The table presented in the paper also helps readers to verify the analysis.

5 Conclusion

This paper investigated the innovativeness of software startups analyzed in the literature. Using a case survey approach, we read 27 published papers and were able to gather 17 software startups that had their products described. Then, we classified them according with the presence or not of market and technology discontinuities. The results highlights that most studied software startups present an innovation with a market discontinuity and without a technical one. Nevertheless, there are also software startups that present a technical discontinuity with or without a market one. This phenomenon could explain why the software startups literature still struggles to define what a software startup is. Besides that, the lack of homogeneity in the studied subjects hinder the application of research results by practitioners and also to compare conclusions from different authors. Such distinction is also important to compare results or import them from other research fields like the new product development or entrepreneurship literature.

A solution to such problems could be a deeper description of studied software startups. Several reasons could prevent researchers to give more details about studied software startups: sometimes founders do not want their idea to be published. Authors can also be limited by the space provided in a conference paper. Another solution could be an explicit commitment from authors that software startups studied in that work are of one kind or a discussion why these differences does not matter to the topic discussed.

References

1. Bajwa, S.S., Wang, X., Duc, A.N., Abrahamsson, P.: How Do Software Startups Pivot? Empirical Results from a Multiple Case Study. vol. 240, pp. 169–176 (2016). https://doi.org/10.1007/978-3-319-40515-5_14
2. Berg, V., Birkeland, J., Nguyen-Duc, A., Pappas, I.O., Jaccheri, L.: Software startup engineering: A systematic mapping study. Journal of Systems and Software **144**, 255–274 (oct 2018). https://doi.org/10.1016/j.jss.2018.06.043
3. Blank, S.: Why the Lean Start-Up changes everything. Harvard Business Review **91**(5), 63–72 (2013)
4. Calantone, R.J., Chan, K., Cui, A.S.: Decomposing product innovativeness and its effects on new product success. Journal of Product Innovation Management **23**(5), 408–421 (2006). https://doi.org/10.1111/j.1540-5885.2006.00213.x
5. Chanin, R., Pompermaier, L., Fraga, K., Sales, A., Prikladnicki, R.: Applying Customer Development for Software Requirements in a Startup Development Program. Proceedings - 2017 IEEE/ACM 1st International Workshop on Software Engineering for Startups, SoftStart 2017 pp. 2–5 (2017). https://doi.org/10.1109/SoftStart.2017.3
6. Duc, A.N., Abrahamsson, P.: Exploring the outsourcing relationship in software startups. In: Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering - EASE'17. pp. 134–143. No. Idi, ACM Press, New York, New York, USA (2017). https://doi.org/10.1145/3084226.3084248

7. Garcia, R., Calantone, R.: A critical look at technological innovation typology and innovativeness terminology: A literature review. Journal of Product Innovation Management **19**(2), 110–132 (2002). https://doi.org/10.1016/S0737-6782(01)00132-1
8. Giardino, C., Wang, X., Abrahamsson, P.: Why early-stage software startups fail: A behavioral framework. In: Lassenius, C., Smolander, K. (eds.) Lecture Notes in Business Information Processing. Lecture Notes in Business Information Processing, vol. 182 LNBIP, pp. 27–41. Springer International Publishing, Cham (2014). https://doi.org/10.1007/978-3-319-08738-2
9. LARSSON, R.: Case survey methodology: quantitative analysis of patterns across case studies. Academy of Management Journal **36**(6), 1515–1546 (dec 1993). https://doi.org/10.2307/256820
10. Marks, G., O'Connor, R.V., Clarke, P.M.: The Impact of Situational Context on the Software Development Process A Case Study of a Highly Innovative Start-up Organization. vol. 770, pp. 455–466 (2017). https://doi.org/10.1007/978-3-319-67383-7_33, http://link.springer.com/10.1007/978-3-319-67383-7 http://link.springer.com/10.1007/978-3-319-67383-7_33
11. McNally, R.C., Cavusgil, E., Calantone, R.J.: Product innovativeness dimensions and their relationships with product advantage, product financial performance, and project protocol. Journal of Product Innovation Management **27**(7), 991–1006 (2010). https://doi.org/10.1111/j.1540-5885.2010.00766.x
12. Nguyen-Duc, A., Seppänen, P., Abrahamsson, P.: Hunter-gatherer cycle: a conceptual model of the evolution of software startups. Proceedings of the 2015 International Conference on Software and System Process - ICSSP 2015 (Idi), 199–203 (2015). https://doi.org/10.1145/2785592.2795368
13. 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. IEEE (aug 2016). https://doi.org/10.1109/SEAA.2016.21
14. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: A systematic mapping study. Information and Software Technology **56**(10), 1200–1218 (apr 2014). https://doi.org/10.1016/j.infsof.2014.04.014
15. Salomo, S., Weise, J., Gemünden, H.G.: NPD planning activities and innovation performance: The mediating role of process management and the moderating effect of product innovativeness. The Journal of Product Innovation Management **24**(4), 285–302 (2007). https://doi.org/10.1111/j.1540-5885.2007.00252.x
16. Sutton, S.M.: The role of process in software start-up. IEEE Software **17**, 33–39 (2000). https://doi.org/10.1109/52.854066
17. Unterkalmsteiner, M., Abrahamsson, P., Nguyen-duc, A., Baltes, G.H., Conboy, K., Dennehy, D., Sweetman, R., Edison, H., Shahid, S., Wang, X., Garbajosa, J., Gorschek, T., Hokkanen, L., Lunesu, I., Marchesi, M., Morgan, L., Selig, C., Oivo, M., Shah, S., Kon, F.: Software Startups - A Research Agenda. e-Informatica Software Engineering Journal **10**(1), 1–28 (2016). https://doi.org/10.5277/e-Inf160105

Why Feature-Based Roadmaps Fail in Rapidly Changing Markets: A Qualitative Survey

Jürgen Münch[1], Stefan Trieflinger[1] and Dominic Lang[2]

[1] Reutlingen University, Alteburgstraße 150, 72762 Reutlingen, Germany
[2] Robert Bosch Smart Home GmbH, Schockenriedstraße 17, 70565 Stuttgart, Germany

Abstract.

Context: Companies in highly dynamic markets struggle increasingly with their ability to plan their future product portfolios and to create reliable feature-driven roadmaps. It seems that the traditional process of product roadmap creation that aims at providing a stable plan for all involved stakeholders does not fulfill its purpose anymore. However, the underlying reasons as well as necessary changes to the roadmap process are not widely analyzed and understood.

Objective: This paper aims at getting an understanding of current problems and challenges with roadmapping processes in companies that are facing volatile markets with innovative products. It also aims at gathering ideas and attempts on how to react to those challenges.

Method: As an initial step towards the objectice a semi-structured expert interview study with a case company in the Smart Home domain was conducted. Four employees from the case company with different roles around product roadmaps have been interviewed and a content analysis of the data has been performed.

Results: The study shows a significant consensus among the interviewees about several major challenges and the necessity to change the traditional roadmapping process and format. The interviewees stated that based on their experience traditional feature-based product roadmaps are increasingly losing their benefits (such as good planning certainty) in volatile environments. Furthermore, the ability to understand customer needs and behaviors has become highly important for creating and adjusting product roadmaps. The interviewees see the need for both, sufficiently stable goals on the roadmap and flexibility with respect to products or features to be developed. To reach this target the interviewees proposed to create roadmaps based on outcome goals instead of product features. In addition, it was proposed to decrease the level of detail of the roadmaps and to emphasize the long-term view. Decisions about which feature to develop should be open as long as possible. Expected benefits of such a new way of product roadmapping are higher user-centricity, a stable overall direction, more flexibility with respect to development decisions, and less breaking of commitments.

Keywords: product management, product roadmap, agile requirements management, requirements engineering, agile development, innovation management, customer development, UX, lean UX, lean development, portfolio roadmap, portfolio management.

1 Introduction

Nowadays the environments for creating new products, services and business models are getting increasingly complex and changing rapidly. Some of the reasons are the emergence of new technologies, high connectivity through the Internet, high availability of knowledge and ressources due to globalization, rapidly changing customer behavior and less predictability of markets and demands. From the point of view of product and service development new development approaches are emerging that are highly customer-centric and data-based with an emphasis on rapid learning. New products and services capture new markets in ever shorter time intervals. New competitors are revolutionizing traditional market structures and require considerable changes from established incubents. This situation has impact on the development and review of product roadmaps. Established enterprises are struggling more and more with their ability to plan their future product portfolios and to create reliable feature-driven roadmaps for the products. Startups also have significant problems with traditional product roadmapping. It seems that the traditional process of product roadmap creation that aims at providing a stable plan for all involved stakeholders does not fulfill its purpose anymore. However, the underlying reasons as well as necessary changes to the roadmap process are not widely analyzed and understood.

This paper aims at understanding current problems and challenges with product roadmapping. It also aims at gathering ideas and attempts on how to react on those challenges. The paper is organized as follows: Section 2 provides an overview of related work. Section 3 presents the research questions and the research study. The results of the study are discussed in Section 4. Finally, an outlook on the future of product roadmaps and further research is sketched.

2 Related Work

A comprehensive overview on the topic of product roadmapping in volatile business environments has been described by Suomalainen et al. [1]. Here, we focus on the core terminology of traditional product roadmapping, describe key problems with traditional roadmaps, and sketch some approaches that go beyond this traditional approach.

Kostoff and Schaller generically define a "road map" as a "layout of paths or routes that exists (or could exist) in some particular geographical space. In everyday life, road maps are used by travelers to decide among alternative routes toward a physical destination. Thus, a road map serves as a traveler's tool that provides essential understanding, proximity, direction, and some degree of certainty in travel planning" [2]. Phaal and Muller consider a roadmap as an aggregation of relevant information to an integrated view on the evolution of a complex system [3]. According to Kappel [4] roadmaps are forecasts of what is possible or likely to happen in order to make better decisions. DeGregorio points out that roadmaps are visualizations of a forecast, which can be applied in a number of key areas such as technology, capability, parameter, feature, product, platform, system, environment or threat and business opportunity [5]. Albright defines roadmaps as living documents that describe a future environment and objectives to be achieved within that environment. In addition, he mentions that roadmaps are plans for how those objects will be accomplished over time. Furthermore, the author suggests that it is advisable to review and update a roadmap over time, otherwise it is not useful [6].

The process to create a roadmap is called roadmapping [2]. Nearly every company applies its own roadmapping process [7]. A main reasons for this is that enterprises have different markets as well as different cultures [8]. An appropriate roadmapping process for a company depends on many factors such as the level of available resources (people, time, budget), the kind of issues being addressed (purpose and scope), or the available information (market and technology). Roadmapping provides a platform for sharing different perspectives and information. Furthermore, the stakeholders of a roadmap can develop a common vision of where the company is going in the future [9].

Roadmapping can be done on different levels. Kappel categorizes roadmaps in four categories based on their purpose and emphasis. These four categories are "Science / Technology Roadmaps", "Industry Roadmaps", "Product-Technology Roadmaps" and "Product Roadmaps" [4]. Phaal et al. identify the following eight types based on their intended purpose: product planning, capability planning, strategic planning, long range planning, knowledge planning, program planning, process planning, and integration planning. In spite of different taxonomies every type of roadmap seek to answer the following questions: 1) Where are we going? 2) Where are we now? 3) How can we get there? [7].

The purpose of a product roadmap is to predict the development of products, features or services over a long period [10]. Typically, product roadmaps are created, reviewed and improved iteratively. For this purpose, human interactions such as face-to-face meetings or workshops play an important role [7].

From the perspective of software product management, the product roadmap provides an overview about the direction of a product, feature or service develoment. Often, a product roadmap provides information about new releases or versions, their schedules and the major topics [11]. Sometimes, a roadmap describes also dependencies between product and platform technology. In some cases, the roadmap contains financial information. For example, estimated revenue and costs are included. In practice, usually the business owner of a product is responsible for the product roadmap. This can include the collaboration and agreement with stakeholders or constant updating of the product roadmap. Usually, a product roadmap has a time horizon of three to five years. [12] In this time frame the roadmap should be undergoing a regular updating process to ensure that the roadmap is developing in the right strategic direction and contains the current state of technical development [1].

Regarding the roadmapping process various approaches have been developed. Lethola et al. [12] suggest that the roadmapping process should consist of the phases "preparation", "approval" and "communication". The phases "theme identification", "core assets" and "roadmap construction" are part of the approach developed by van de Weerd [13]. Vähäniitty [14] considers the process in four steps, which should be performed periodically in order to adjust the roadmap to the changing market situations including new information. The steps are defined as "define strategic mission and vision", "scan the environment", "revise and distill the product vision as product roadmaps" and "estimate product life cycle and evaluate the mix of development efforts planned". Each step has defined objectives. The process is especially developed for creating and updating product roadmaps.

Komssi et al. [15] suggest a six-step roadmapping process based of the analysis of the customer value and customer´s processes. The approach includes the building of a cross-functional team (first phase), the examination of the business strategy (second phase), the selection of a customer segment (third phase), the identification (fourth phase) and analysis (fifth phase) of customer activities and linking the business potential of customer activities into the roadmap (sixth phase).

According to the study "Roadmapping" [16], roadmaps are widely developed, distributed and used in a feature-driven mode. This means that the roadmap contains products or features for a defined time horizon.

In the following, several reasons for using traditional roadmaps and problems with traditional roadmaps are summarized (based on Cagan [17]). Important reasons for using traditional feature-based roadmaps are that the management of a company wants to make sure that the teams are working on the highest-business-value items first. On the other hand, the management wants to be able to predict, when the products or features are ready for market launch. In order to do this, the management usually arranges a quarterly or annual planning session, where the leaders consider the ideas and negotiate a product roadmap. This procedure implies multiple challenges which will be discussed in the following.

First of all, a feature-driven roadmap is only a scheduled list of product or features. In rapidly changing environments such a roadmap includes many uncertainties and is typically undergoing many changes over time. Consequently, a company might lose reliability to external partners and the management might lose an essential controlling tool.

Another issue due to Cagan is that anytime you put a lot of ideas on a document entitled roadmap, no matter how many disclaimers you put on it, people across the company will interpret this item as a commitment to develop it. This leads to a change of focus from the actual needs of the customer to the functionality of the product or the system with its features. The criteria for success is no longer customer satisfaction, but to deliver them "on time". This procedure leads to the risk that the enterprise moves in the wrong direction and in some cases might run out of business.

Furthermore, Cagan mentions that at least half of the ideas on a product roadmap are just not going to work. The most frequent reasons are that the customers are not excited about an idea. This circumstance can be attributed to the underlying assumptions about the user or the feature itself. Here is an example: an assumption could be that the user would like to have an intelligent roller shutter control for the summer. However, the real customer might only need a cool room. Therefore, the assumption that the intelligent roller shutter control is the right solution for this customer is not necessarily correct. If there is a better solution available for the customer, the product "intelligent roller shutter" might not be able to survive in the market.

Several approaches on how to evolve traditional roadmapping have been proposed. Pichler [18] distinguishes between a so-called feature-based roadmap and a so-called goal-oriented product roadmap. The feature-based roadmap can be seen as the format that is traditionally used for product roadmaps. It defines the dates for upcoming releases and the features that are included in each release. It does not define corresponding goals that are expected to be fulfilled with each release. In contrast the goal-oriented roadmap includes the following information: the release dates, a goal associated with reach release, and the features associated with a release. Figure 1 shows the difference between these two types of product roadmaps. The goal-oriented product roadmap shifts the conversation from discussing features to agreeing on strategic objectives, making smart investment decisions, and aligning stakeholders [19]. Goal-oriented roadmaps do not consider explicitly if certain features on the roadmap are suitable means for reaching the respective goals.

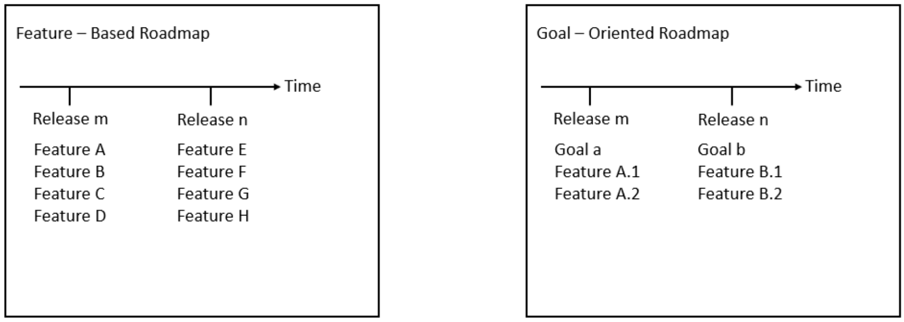


Fig 1. Feature-based Roadmap vs. Goal-oriented Roadmap [18]

Jeff Patton has created an approach called User Story Mapping. It starts with the identification of the customer journey along the horizontal axis. In the case of a web shop, the customer journey could be “search for a product”, “view product details”, “add a product to shopping card”, and “buy the product”. In a second phase the core user stories are determined, prioritized, and mapped to the customer journey. Examples for user stories are “enter credit card info”, “enter delivery address”, and “confirm order”. In the further phases the definition of the releases take place [20]. An interesting aspect of User Story Mapping is that releases can be planned by walking down the vertical axis and defining goals. Appropriate functionalities can be considered and tested before implementation with respect to reaching the release goal. This way, User Story Mapping can be seen as a new way of product roadmapping that goes beyond traditional formats and approaches.

3 Study Approach

This section gives an overview of the study approach. It starts with presenting the research question and continues with a description of the study context, i.e., the case company. Afterwards, the study design including the data collection and analysis, the study execution, and the discussion of validity are sketched.

For the study the following research questions are defined:

- RQ1: Which approaches and methods for creating and updating a product roadmap are currently applied by the case company?
- RQ2: What challenges and success factors are associated with product roadmapping in the case company?

3.1 The Case Company

The study has been conducted at the Robert Bosch Smart Home GmbH (HOME), referred to as case company in the following. The case company is a business unit of the BOSCH Group. It was founded in 2016 as an independent subsidiary. It is engaged in smart home activities and offers a wide range of products, features and services in the business field of smart home. Products developed by HOME are, for instance, intelligent heating control and automated house surveillance. The actual number of employees is about 150. For this study interviews with four employees from the case company were conducted who were involved in the roadmapping process [21].

3.2 Study Design

The study was conducted by using a qualitative survey method. The qualitative survey method was chosen because the study has the objective to obtain new insights with respect to procedures, challenges and success factors in the area of "product roadmapping" in the context of a case company. To achieve this objective, the experience, opinions and views of the experts needed to be obtained. Therefore, the qualitative survey method (including semi-structured interview, observation, and content analysis) was preferred over the quantitative survey method [22].

Moreover, Fink identifies several opportunities, in which a qualitative survey method is appropriate. The following four aspects are relevant regarding this study: 1) The study is focused on investigating the knowledge and opinions of experts in a particular field. 2) The study intends to collect information in the interviews with own words rather than with using predefined choices. 3) There is not enough prior information of the study subject to enable either the use of standardized measures of the construction or a formal questionnaire. 4) The sample size is limited due to access or resource constraints [23].

3.3 Data Collection and Analysis

Semi-structured expert interviews with participants of the case company were used to collect data. The expert interview is a method of qualitative social research. [24] In an expert interview the participants can answer the questions by using free speech and a self-chosen terminology. In the following, typical characteristics of an expert interview are listed.

Table 1. Characteristics of an expert interview [25]

Motivation	**Professional interest**
Process:	Constructive
Motivation of the interviewee:	Presentation/Transfer of knowledge
Criteria of exclusion (interviewee):	Interviewee is not an expert
Criteria for exclusion (interviewer):	Unfamiliarity with the topic

An interview guide was developed to structure and focus the interview with the predefined topics and to ensure the thematic comparability of the various interviews (the complete interview guide is available in Appendix 1). In addition, the interview guide was created in order to avoid that important aspects are ignored [26].

The developed interview guide consists of three parts. It begins with an opening part including the background of the interviewed person. The main part contains questions with respect to the predefined topics. Finally, the closing part considers topics which were not considered up-front in the interview guide [27].

For a detailed data analysis, all interviews were audio recorded and transcribed. The most important findings were identified and examined through a analytic content analysis.

3.4 Study Execution

The study participants were selected experts from the case company. According to Mieg [25] the experts can be characterized as persons who have authorisation to a certain field and are involved in decision making processes based on their position. In this research the authors refer to those experts, who have specified knowledge and skills about product roadmapping and are involved in roadmapping activities.

The case company was represented by four interviewees. All interviewees held positions in the middle management. The participants represented the departments sales business operations, IT coordination, product management and brand and marketing communications. The purpose and the procedure of the study were shared with the interviewees via an up-front email.

The individual expert interviews were conducted in the office at the case company on September 21, 2018. The average length of the interviews was 47 minutes, with the range spanning between 33 and 52 minutes. One researcher conducted all interviews in face-to-face conversations. An overview of the background of the interviewees is shown in Table 2. The experience refers to the amount of years in which the person was involved in roadmapping activities.

Table 2. Overview of the interviewees

Interviewee	Role	Experience
Interviewee 1	Head of Sales Business Operations Department	20 years
Interviewee 2	IT Coordinator	1 year
Interviewee 3	Head of Product Management Department	12 years
Interviewee 4	Marketing and Brand Manager	20 years

3.5 Validity

Yin [28] proposes to consider the construct validity, the internal validity, the external validity, and the reliability for assessing the validity and trustworthiness. We use this framework as the basis for the discussion of validity of our study. Other frameorks exist such as the framework from Campell and Stanley [29] that are also applicape for this kind of studies.

Construct validity refers to the correct operational measures for the concepts being studied [28]. As a means for establishing construct validity the goal and the purpose of the interviews were explained to the interviewees before the interviews. In addition, the way of data collection through semi-structured interviews allowed for asking clarifying questions and avoiding misunderstandings.

Internal validity refers specifically to whether an experimental treatment/condition makes a difference or not, and whether there is sufficient evidence to support the claim [29]. This criterion can be tested with respect to the validity claims for communicative actions, according to Habermas [30]. These criteria are defined as follows: 1) Clarity describes to which extent the interviewees understand the questions or whether there occur any linguistic discrepancies; 2) Legitimacy refers to the cooperativeness of the interviewees; 3) Trueness refers to find no contradictions in the statements, 4) Sincerity consider the completeness of the statements. The following discusses the internal validity according to Habermas:

- Clarity: The interviewees were experts with many years of experience in the field of roadmapping. Each participant was a native speaker in the interview language German. In cases where the questions were unclear to the participants, they asked further questions.
- Legitimacy: Each interviewees were interested in the research and answered the questions in a detail manner. So, in summary there was a very cooperative atmosphere.
- Trueness: The experts came from different disciplines, so they asked the questions from various perspectives. The analysis showed that there were no major contradictions between the perspectives.
- Sincerity: Each interviewee answers the question extensively and there was no indication of missing parts of the topic.

The **external validity** is defining the domain to which the studies can be generalized [28]. Regarding this study the external validity is restricted, because the results are only valid in the context of the case company. Thus, the results are not transferable to other fields of investigation. Anyhow, the company might be similar to other German companies in the IoT or Smart Home domain. Therefore, an analytic generalization might be possible to such similar companies.

The **reliability** describes whether a study produces stable and consistent results. For example, the data collection procedures can be repeated with the same results [28]. The reliability was supported by providing an interview guide that is publicly available. Although the study was just an initial effort to answer the research questions, the analysis has been conducted in a systematic and repeatable way. Therefore, a replication of the study and a reduction of researcher bias is supported.

4 Results

This section sketches the product roadmapping practices of the case company (answering research question RQ1). Afterwards the challenges and the success factors that were seen in the case company are outlined in two different sections (answering research question RQ2).

4.1 Product Roadmapping Practice

The current product roadmap format of the case company resembles a coordinate system. On the y-axis you find domains like security, climate or lighting. The x-axis represents the time dimension (see Figure 2). Usually a time horizon of 12 months is used. The products and features are put on the roadmap according to their associated domain and their planned development time (i.e., start and end date). Moreover, each feature contains the information when the rollout (i.e., the software deployment to the customer) is ready or in the case of hardware when the market launch is to be expected. This procedure provides a clear overview of the planned market launches to external and internal partners.

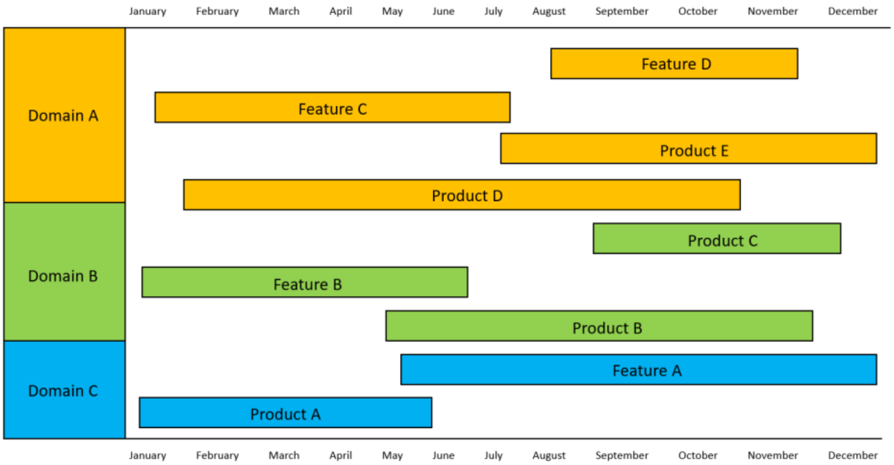


Fig. 1. Product roadmap format at the case company

Currently the management board is responsible for the product roadmap. However, the management is delegating the product roadmap creation into the hands of the product management. In practice, the head of the product management department is responsible for the product roadmap. This responsibility includes creating and updating the product roadmap as well as the coordination of other stakeholders with respect to the roadmap. These stakeholders are the departments “Portfolio Management”, “Engineering”, and “Marketing and Sales”.

For creating the product roadmap and for adding new products, features or services the following approach is applied: *“Currenty we have the procedure that the management and I, the head of the product management department define criteria to asses a product, service, or feature proposal. Typical examples for such criteria are 'Does the product have a unique selling proposition?', 'Is there a demand from the perspective of the customer?', and 'How much revenue is estimated?'. Each of these criteria is given a specific weight. This could be, for example, a factor 4 for the estimated revenue while the customer demand might be calculated with a factor 3. Every product, feature or service is then evaluated and receives a score based on the mentioned criteria. This score reflects a priority, i.e., the higher the score the higher is the priority and the product, feature or service is more likely to be put onto the roadmap at an earlier time.*” (Head of Product Management Department). Furthermore, an analysis of the social media channels and service-tickets is conducted. The results of these analyses are also included in the decision process and can lead to minor adjustments.

New ideas for products, services or features can stem from many different sources, especially customers. In case of an update for an existing product, surveys with users will be conducted.

Every month, the product management has a meeting with the other stakeholders in order to make a concrete decision about which products, services, and features should be put on the roadmap. At this meeting, the product managers are presenting their findings of their market research and discuss them with the other stakeholders. The market research is conducted by the department "Product Management" and includes which products or features are expected from customers and what products are developed by the competitors. Another input to the meeting is so-called GfK data. This GfK data describes consumer behavior and can be used to identify potential delivery areas.

In the next step the development department estimates the development time. The estimation takes the budget and the available resource into account. The estimation also aims at answering the question whether a completion of the planned feature is possible in the scope of the target release. If it is not possible to deliver on time, the product, feature or service might be moved to a later release.

The prioritization of the previously selected products, features or services is mainly based on financial aspects. *"A financial forecast is conducted with the goal to find the products, features or services that have the highest impact on achieving the revenue targets of the company. This financial forecast has the highest impact on the prioritization*." (Head of Product Management Department) Other criteria that are used in the prioritization are the strategic alignment, the customer demand as well as the contribution to the development of a competitive advantage.

Another topic of the monthly meeting is the revisiting of the current roadmap. The participants analyze the impacts of the last four weeks and try to identify deviations from the roadmap or needs for changing the roadmap. A typical situation is a change of capacity or budget. Such a situation might be that the company cannot develop a planned hardware because of a lack of budget. Another example is that the engineering has to fix a lot of bugs the next two sprints. This might lead to a delay in the completion of the planned features and to a deferral of the products on the product roadmap. Consequences could be that features with a low prioritization are removed from the product roadmap or a market launch gets delayed. Also market-driven events (e.g., from DIY stores and electronic stores) or technological innovations might lead to a change of the product roadmap. *"The rise of conversational interfaces such as Amazon's Alexa is an example for a technological innovation that has a significant impact on many product roadmaps in the smart home domain. Without the integration of such devices or ecosystems, the competitiveness of many smart home products would be threatened*." (Head of Sales Business Operation Department). The revisiting of the product roadmap includes a review with respect to delays of prioritized products.

4.2 Challenges of the Product Roadmapping Process

The case company operates in an innovative and highly dynamic market environment with rapid changes and disruptive participants entering the market. This imposes several challenges to the roadmapping process. Table 3 gives an overview of the challenges that were mentioned in this study.

The product roadmap developed by the case company covers a 12 months period. Thus, *[…] concrete products or features are defined over an incalculable long time horizon with many uncertainties […] Nowadays, a long-term-planning with reliable and stable information (i.e., with features, products and services) is no longer possible* *due to rapidly changing markets."* (IT Coordinator) The volatile market environment and difficulties with predicting development activities require frequent updates of the product roadmap. Reasons for these changes are, for instance, a decline in demand for certain products, development delays due to unforseen events or other important things. Frequent changes to the roadmap currently lead to high additional cost and sometimes delayed marked launches of new products.

Furthermore, a constantly changing roadmap is likely to decrease the employee´s awareness for the overall strategy and company vision. Moreover, the new planning consumes a lot of capacity of the participating employees which could be used more efficiently.

"One factor for creating a roadmap is that marketing needs to plan campaigns long-term ahead and sales requires an reliable outlook of the product portfolio including the future products and features to present it to potential customers." (IT Coordinator) In both cases the mentioned departments require a certain reliability to which point in time a product, feature or service will be available.

Finally,*"in some cases ideas for new products, services or features come from management or investors with the expectation that these ideas will be implemented without any delay and independently from the current planning. Often, the implementation of these ideas leads to an unforeseen change of the product roadmap."* (Head of Product Management Department) The result is a shift of some features to a later point in time and hence often means a delayed product launch.

Table 3. Challenges with current product roadmapping

Product Roadmapping – Current Challenges
Many uncertainties exist due to rapid changes of markets, technologies, and customer behaviors.
Time horizon of a roadmap is too long.
Frequent changes of the current roadmap are necessary.
Frequent changes of the roadmap impose severe consequences (high cost, delays, and planning overhead).
Difficult alignment of the roadmap with product vision and long-term company strategy.
Marketing and sales require long-term predictions for features, products and services in order to plan their activities (such as campaigns).
Management or investors sometimes overrule product roadmaps.

4.3 Success Factors of Product Roadmapping

Another objective of this study was to gain insights into the success factors of product roadmapping. Table 4 gives an overview of the expected success factors for future roadmapping activities that were mentioned in this study.

The experts from the case company mentioned that a good understanding regarding the market as well as the ability to live with uncertainties are important success factors. *"A good understanding of the market is necessary for creating a good roadmap. Maybe* *also the abibility to deal with uncertainties is necessary. This means accepting that nobody exactly knows which product we will launch in a year […] and also accepting that the roadmap will become fuzzy looking in the long term horizon."* (IT Coordinator) This means that each employee must accept that a roadmap provides detailed information only over a short period of time (e.g., product planning for the next 3 months). In the case of a volatile environment with rapid changes it is impossible to plan in detail for a long-time-period. The planning should be conducted continuously to ensure that the roadmap is always up to date and that the company can always rely on a detailed plan for a short-term period.

The experts also mentioned that a roadmap should help to give all stakeholders an idea of the product vision and the direction the company will go in the future.

Another central theme that was mentioned as success factor in the interviews is that the needs of the customers should be included in the roadmapping process, […] *"the fulfilment of customer needs is the prerequisite for creating successful products that generate revenue."* (Head of the Product Management Department) A central question has to be: *"In which way do the contents of the roadmap contribute to solving a current problem of the customer?"*(IT Coordinator) The financial review was also mentioned as a success factor.

Table 4. Success factors for product roadmapping

Product Roadmapping – Success Factors
Ability to live with uncertainty.
Good understanding of markets and customer behaviors.
Detailed planning only for a short period of time.
Continuous planning.
Connecting the roadmap to the fulfillment of customer needs and business goals.
Alignment of the roadmap with product vision and company strategy.

5 Outlook and Further Research

As part of the study we also asked the participants about their proposals for improving product roadmaps and related process in the future. We also aim at building a substantive theory of goal-driven or outcome-driven roadmaps that can be applicable in a wider context. We expect that we will also narrow down research questions when gaining a better understanding of the research area.

The interviewees mentions that future roadmaps should be structured in a problem- or outcome driven form. This means that the roadmap should not contain products, features or services, but instead current needs and problems from the perspective of the customers (i.e., customer outcomes) and the related business goals (i.e., business outcomes). Thus the roadmaps are widely outcome-oriented and the way to reach this outcome (e.g., which features are built to solve the customer problem and/or reach the business outcome) is left open. This procedure allows that all aspects such as future technologies and trends can be taken into consideration. It also allows to conduct experiments in order to determine if certain features are suitable to reach the outcomes (ideally before implementing them).

Therefore, future product roadmap should be designed in an outcome-oriented way. This means that the information contained in a long-term roadmap should only reflect the current needs or problems of the customers as well as the business goals and not the possible solutions. This allows the company to stay flexible in deciding which solution fits best and therefore leads to a better fulfillment of the customer demands. Moreover, it is assumed that an outcome-driven and user-centric approach for the roadmapping process offers an effective planning of the operative measures and more space for creativity.

In summary, the traditional procedure for the roadmapping process is not suitable anymore for an agile and innovative environment. Hence a new approach is required. This new approach has to provide a flexible customizability to adapt to rapid market changes as well as provide sufficient planning security with respect to outcomes. Other disciplines such as marketing and sales will also need to change their way of working. It might be that they need to plan long-term marketing campaigns based on outcomes instead of available features.

The challenge for many companies will be to adjust and replace their traditional product roadmaping and introduce a new modern roadmapping process that makes them ready for a volatile highly dynamic environment. Further investigations regarding the abilities of an outcome-driven and user centric roadmapping process (including the roadmap format and organizational and cultural aspects) are necessary in order to find a new approach that fits today's dynamic and complex environments.

References

1. Suomalainen, T: Changing the planning for agile and lean software development – From roadmapping to continuous planning. Juvenes Print, Tampere (2016).
2. Kostoff, R., Schaller R.: Science and Technology Roadmaps. IEEE Transactions on Engineering Management 48(2), 132 – 143 (2001).
3. Phaal, R., Muller, G.: An architectural framework for roadmapping: towards visual strategy. Technological Forecasting and social change – An international journal 76(1), 39 – 49 (2009).
4. Kappel, T.: Perspectives on roadmaps: how organizations talk about the future. The Journal of Product Innovation Management 18(1), 39 – 50 (2001).
5. DeGregorio G.: Technology Management via a set of dynamically linked roadmaps. In: Proceedings of the 2000 IEEE Engineering Management Society. EMS-2000 (Cat. No.00CH37139)", pp.184 – 190. IEEE, Albuquerque, NM, USA (2002).
6. Albright, E.: A Unifying Architecture for Roadmaps Frames a value scorecard. In: IEMC '03 Proceedings. Managing Technologically Driven Organizations: The Human Side of Innovation and Change, pp. 383 – 386. IEEE, Albany, NY, USA (2004).
7. Phaal, R., Farrukh, C.J.P., Probert D. R.: Developing a technology roadmapping system. In: A Unifying Discipline for Melting the Boundaries Technology Management. pp. 99 – 11. IEEE, Portland, OR, USA, (2005).

8. Groenveld, P.: Roadmapping Integrates Business and Technology. Research Technology Management 50(6), 49 – 58 (2007).
9. Phaal, R., Farrukh, C.J.P., Probert D. R.: Technology roadmapping - A planning framework for evolution and revolution. Technological Forecasting and Social Change 71(1-2), 5 – 26 (2004).
10. Albright, R. E., Kappel T. A.: Roadmapping In the Corporation. IEEE Engineering Management Review 31(3), 31 – 40 (2016).
11. Kittlaus, H.B, Clough, P.N: Software Product Management and Pricing: Key Success Factors for Software Organizations. Springer-Verlag, Berlin Heidelberg (2009).
12. Lehtola, L., Kauppinen, M. and Kujala. Linking the Business View to Requirements Engineering: Long-Term Product Planning by Roadmapping. In:
Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE), pp. 439 – 446. IEEE, Paris (2005).
13. Van de Weerd, I., Bekkers, W., Brinkkemper Developing a Maturity Matrix for Software Product Management. In: Tyrväinen, P., Jansen, S., Cusumano M. A. (eds.) Software Business – First International Conference, ICSOB, pp. 76-89. Springer Heidelberg (2010).
14. Vähäniitty, J., Lassenius, C., Rautiainen, K.: An Approach to Product Roadmapping in Small Software Product Businesses. In: Kontio J., Conradi R. (eds.) Proceedings of the 7 th European Conference on Software Quality (ESCQ) - Quality Connection, pp. 12-13. Springer, Heidelberg (2002).
15. Kommsi M., Kauppinen, M., Tohonen, H., Lethola, L., Davis, A.M.: Integrating Analysis of Customer´s Process into Roadmapping - The Value-Creation Perspective. In:19th International Requirements Engineering Conference (RE), pp. 57 - 66. IEEE, Trento, Italy (2011).
16. Schimpf, S., Abele, T.: Praxisstudie Roadmapping, Frauenhofer IAO und Tim Consulting (2016).
17. Cagan, M.: Inspired: How To Create Products Customer Love. 2 nd edn. SVPG Press, California (2018).
18. Pichler R.: “Strategize: Product Strategy and Product Roadmap Practices for the Digital Age. Pichler Consulting (2016).
19. Pichler R.: Agile Product Management with Scrum: Creating Products that Customer Love. Upper Saddle River, New Jersey (2010).
20. Patton, J.: User Story Mapping: Discover the Whole Story, Build the Right Product. O'Reilly Media, Sebastopol (2014).
21. Bosch Smart Home GmbH, internal source (2018).
22. Thomas, T. R.: Blending Qualitative & Quantitative Research Methods in Theses and Dissertations, Corwin Press, Thousand Oaks, California (2003).
23. Fink, A. Analysis on Qualitative Surveys. In: Fink A. (eds.) The Survey Handbook, pp. 61 – 78. SAGE Publications, Thousand Oaks, California (2003).
24. Ullrich, P.: Das explorative ExpertInneninterview: Modifikationen und konkrete Umsetzung der Auswertung von ExpertInneninterviews nach Meuser / Nagel. In: Engartner, T., Kuring D., Teubl T. (eds.) Die Transformation des Politischen. Analysen, Deutungen, Perspektiven, pp. 100 – 109, Karl Dietz Verlag, Berlin (2007).
25. Mieg, H. A., Näf M.: Experteninterviews in den Umwelt und Planungswissenschaften – Eine Einführung und Anleitung, 2nd edn. Institut für Mensch-Umwelt-Systeme (HES), ETH Zürich (2005).
26. Meuser, M., Nagel, U.: Experteninterviews – vielfach erprobt, wenig bedacht: ein Beitrag zur qualitativen Methodendiskussion. In: Garz, D., Kraimer, K. (eds.) Qualitativ-empirische Sozialforschung: Konzepte, Methoden, Analysen, Westdt. Verl., Opladen (1991).

27. Mayer, H.O.: Interview und schriftliche Befragung. Entwicklung, Durchführung und Auswertung, 4nd edn. Wissenschaftsverlag GmbH, München (2009).
28. Yin, R. K.: Case study research: Design and methods, 5. edn. SAGE Publications Inc., London (2014).
29. Campbell, D., Stanley J.: Experimental and quasi-experimental designs for research, Houghton Mifflin Company, Chicago (1963).
30. Habermas, Jürgen: The Theory of Communication Action – Reason and the Rationalization of Society, edn. 1 Beacon Press, Boston (1984).

Appendix 1. Interview Guide

Background of interviewee:

1. What is your current position in your company?
2. How many years have you been working in the company?
3. How long are you involved in the topic product roadmapping?

Company Information:

4. Can you briefly describe the business sector your company operates in and the products it develops?
5. What kind of development process do you use?
6. How often do you deploy new versions to customers?

Current roadmapping practices:

7. Who is responsible for the development of the product roadmap in your company?
8. Which information does the product roadmap contain?
9. What is the procedure of product roadmap creation?
10. Who is involved in the product roadmapping process?
11. Which information is used for creating the product roadmap? Where does this information come from?
12. How do you prioritize the product roadmap?
13. How do you make decisions which contents are included or removed from the product roadmap?
14. How do you review the product roadmap?
15. What are criteria for a good product roadmap?
16. In which way do you integrate other stakeholders such as other departments, customers, or suppliers in the product roadmapping process?
17. In which situations are you changing the product roadmap and how do you change it?
18. What is the process for changing the product roadmap?

Challenges, success factors and improvement proposals:

19. Are there any challenges or obstacles regarding the product roadmap process?
20. In your opinion, which factors are supporting the product roadmapping process?
21. Do you think your current practices of product roadmapping are ideal? If not: How should they ideally be performed in the future?

Final questions:

22. Do you have any further comments about product roadmapping issues in the context of your company?
23. Do you have any further questions related to this interview or the study in general?

Software Startup Education Around the World: A Preliminary Analysis*

Rafael Chanin[1], Dron Khanna[2], Kai-Kristian Kemell[3], Wang Xiaofeng[2], Afonso Sales[1], Rafael Prikladnicki[1], and Pekka Abrahamsson[3]

[1] PUCRS, Porto Alegre, Brazil
{rafael.chanin, afonso.sales, rafaelp}@pucrs.br
[2] Free University of Bozen-Bolzano, Bolzano, Italy
{dron.khanna, xiaofeng.wang}@unibz.it
[3] University of Jyväskylä, Jyväskylä, Finland
{kai-kristian.o.kemell,pekka.abrahamsson}@jyu.fi

Abstract. New software startups are born everyday around the world. Nonetheless, failure is the fate of most of them. The community already knows that several facts, such as market competition or lack of resources, can impact the destiny of a startup. However, little has been explored in term of the impact of software startup education on the success of failure of startups. In this sense, this study presents the initial steps that we are taking to understand how software startups are taught around the world. To do so, we design a qualitative survey aimed at software startup educators at Universities. Our goal is to gather enough information so we can help the academic community in improving their own courses. So far, we have gathered 10 responses from lecturers across the globe. This paper describes these findings.

Keywords: Software Startup Education · Software Startup · Entrepreneurship.

1 Introduction

In the last decade, we have witnessed significant advances in technology, specially after the popularization of the Internet. Today, any person with software development skills is able to create applications that can be reached by millions (and even billions) of people [11]. Companies such as Google, Netflix, and WhatsApp are examples of organizations that were born under these conditions.

These technology endeavours, that are developed under high uncertainty, are called *startups* [2]. Most startups follow the *lean startup* methodology [19], which combines short software development cycles with constant interaction with users. The goal is to reduce risk by focusing on constant learning [4]. A startup needs to find a business model as quickly as possible, otherwise it may run out of resources before turning itself into a company. Therefore, a startup must focus

* This work is partially funded by FAPERGS (17/2551-0001/205-4).

on understanding what customers need, what they expect, and how much they are willing to pay for a solution [6].

From an education perspective, several software engineering/computer science courses have focused on entrepreneurship in the last years ([8, 12, 15]). In addition, several technology-related courses are adapting their curriculum in order to include startup/entrepreneurship content [9].

One of the biggest challenges reported on these studies is the lack of a realistic environment for student to work on their startups [18]. Since the main goal of a startup is to solve real customers' problems, faculty must find ways to provide real challenges to students.

In this sense, the goal of this paper is to understand how software startup is taught by lecturers/professor across the world. So far, we do not know how courses are carried out aside from papers describing individual experiences. Therefore, this study focuses on the following research questions:

- **RQ1: Aside from conventional lecture-based courses ending in an exam, how are software startups taught in universities?**
- **RQ2: How do these courses deal with the multidisciplinary nature of a software startup (business, technology, design)?**

The remainder of this paper is organized as follows. Section 2 presents the related work. Section 3 describes the research method used in this research. In Section 4 we show our preliminary results. Finally, we draw our conclusions in Section 5.

2 Background

In this Section we define and explain in details what a software startup is. In addition, we depict how software startup education is being explored by the academic community.

2.1 Software Startups

Though software startups have recently had a large economic impact across the globe, no clear consensus on what exactly a software startup is exists [22]. Startups are not simply small, new companies seeking to grow into larger corporations, nor is there a clear point after which a startup has clearly grown into a mature company. Despite the lack of a consensus on an exact definition, some shared understanding of characteristics that define startups does exist.

Startups operate under a lack of resources, both in terms of time, manpower, and finances [17, 21]. They largely rely on external funding especially early on in their lifecycles, and have little to no prior operating history [1, 21]. Though not every single startup is a software startup or even focused on technology-based products, startups by definition are often nonetheless considered to be software or more generally tech companies, especially by practitioners [1]. Software startups specifically, however, operate in particularly volatile markets, using current top-of-the-line technologies to engineer innovative products and services [13]. This, combined with the scarcity of resources, leads to software startups generally operating under highly uncertain conditions [20].

Perhaps the most important difference between a conventional small business and a startup is that startups are characterized by clear intentions for high growth. While small companies generally wish to grow, and will usually do so if presented with a clear opportunity, startups are founded with plans for high potential growth from their inception. Indeed, startups typically seek a particularly highly scalable business model [1]. In the case of software startups in particular, this is further highlighted by the digital nature of software: digital goods are easily distributed or sold world-wide.

Another important characteristic of a startup is that startups are temporary: a startup does not want to keep being a startup. A startup will either fail somewhere along the way or grow into a mature organization. Though it is unclear when exactly a startup ceases to be a startup, drawing from the definition of Blank [1], one could argue that a startup ceases to be a startup when it has found the highly scalable and sustainable business model it sought.

For the purpose of this study, we consider startups to be temporary organizations seeking a highly scalable business model. Software startups, on the other hand, we consider startups that deliver value primarily through software. For instance, though Uber is a taxi company, it nonetheless delivers its value to its customer through the software used to access the service; after all, it does not actually own a single taxi.

Software startups are typically associated with success stories such as that of the aforementioned Uber. However, the majority of software startups fail [7], with some estimates citing numbers as high as 90%. Despite their high rate of failure, software startups have had a notable impact on the economies of more developed countries, especially in the last decade [22]. As a result of recent technological advances, an average supermarket laptop can now be used to develop software which can then be hosted in the cloud, whereas twenty years ago the cost of developing and distributing software was much higher. This sharp decrease in required resources in software development has resulted in an increasing number of software startups.

As software startups have become more numerous and increasingly impactful at an international economic level [22], they have also become increasingly relevant from the point of view of education. It is not uncommon for software engineering students to involve themselves in software startups both during their studies and after graduation. In fact, software startup practitioners in general tend to be inexperienced [21, 14]. Just as entrepreneurship in general is taught in educational institutes across the globe, startups as one of its facets have grown prominent enough to warrant unique focus. As established in this section, startups differ from conventional small companies, making generic entrepreneurship education not fully applicable to them.

In terms of business, whereas founding a conventional company would see one write a detailed business plan for investors and perhaps take out a loan cover material costs as well, startups prefer one-page-long business model canvases over business plans and are far more focused on acquiring outside funding through short public talks referred to as pitches. Though startups are not completely unlike conventional small businesses at their core, startup entrepreneurship has grown into a sub-culture with its own community and jargon. Startup events across the globe (for instance, Slush [4]) attract famous practitioner speakers and large, successful startups are motivational success stories for up-and-coming startup practitioners. Startup incubator organizations and various startup-related societies support startups during various stages of their lifecycles. As a result, startup companies use constructs that differ from conventional business vocabulary and have their own practices, for instance, in terms of searching for investments.

More specifically in relation to software startups, software startups have been shown to develop software differently from SMEs and large corporations [17]. Software startups often use varying agile methods and practices, or even develop software purely ad hoc [17]. Similarly, software startups are characterized by particularly high levels of technical debt. As time-to-market is essential and the lack of resources forces software startups to develop quickly, software startups find themselves accumulating technical debt. After all, in the case of failure, which is the fate of most software startups [22], that technical debt will never have to be addressed.

Just as organizations such as startup incubators and various startup event organizations have sprouted to support the high number of emerging software startups, some scholars have also begun to devise and carry out startup-related university courses. Whereas business and entrepreneurial education in general has a long-standing history in the academia, startup and software startup education as its subset is still in its infancy. In the following sub-section, we discuss the current state of practice on software startup-related education based on literature.

2.2 Software Startup Education

Three of the authors of this papers have worked on a systematic mapping review on software startup education [5]. The goal of this work was to identify the main academic contributions on software engineering education in the software startup context by understanding the state-of-the-art research on software startup development education, and by collecting best practices and methodologies used on software startup education. After running the systematic mapping process, the researchers ended up fully exploring 31 publications. In this section we summarize the main contributions from this work.

The authors broke down the research into two research questions. The first one was related to tools, models, methodologies, and frameworks used in a software startup education context, whereas the second research question focused on best practices and lessons learned.

[4] https://www.slush.org

Regarding the first question, the authors found that the main components used to teach software startups are:

- *Business Model Canvas* - the Business Model Canvas [16] helps students understand startups in its entirety. Since technology students tend to focus more on the product and not on other important aspects of a startup (such as the market), the Business Model Canvas provides a way open students' mind in order for them to envision the big picture;
- *Customer Development Process* - The Customer Development Process, proposed by Blank and Dorf [2], helps students validate their business hypothesis. By telling entrepreneurs to "*get out of the building*", Blank and Dorf are saying that the validation process goes beyond product development; it is necessary to get closer to real potential customers in order to understand their needs;
- *Design Thinking* - The Design Thinking process is mostly used for ideation, specially when students need to come up with an idea to work on, or when they need creative solutions to move one with their projects; *Agile* - Whenever students need to create a real software project, Agile is the most used methodology. Since Agile methods, such as Scrum, provides flexibility and are receptive to project/product changes, it fits well on a software startup context.

In addition, some studies brought interesting insights for those who involved in software startup courses. For instance, Génova and González [10] claim that students must go through three stages in order to achieve a complete software startup education: (i) *instruction*, (ii) *training*, and (iii) *mentoring*. The first stage is related to tradicional educational settings, when students are able to learn content and are assessed by exams. The second stage is when students are able to choose their own way to solve a problem. Finally, the third stage is when students are able to self-propose their own goals and objectives.

In another study, Buffardi *et al.* [3] argue that working with mock-up projects is not effective, since students do not experience real life challenges. On the other hand, it is hard to emulate or to work with real world projects in an academic setting. The proposed solution was to create a multidisciplinary course with both software engineering and business students. In this situation, business students act as customers. Even though this is not an ideal scenario, at least provides a good overview of a software startup development process.

The conclusion regarding this first research question is that there are several different approaches being used. Since courses have different goals, objetives, and resources, each one ended up having a different focus. For instance, some courses just aim at inspiring students to further pursue an entrepreneurial career, while others focus more on technical aspects of a software startup.

In regards to the second research questions, we can break down the learnings into four categories:

- Teaching: The journey is more important that the endpoint. In this sense, lecturers should assess students' progress. Exams are not a good option since concepts are easy in theory, but very hard to be applied in practice;

- Real Projects: Whenever possible, courses should be connected to the market. When students work with real projects, their engagement and excitement rises. However, it is not always possible to do so. If that is the case, faculty should provide means for students to at least mimic a real world scenario;
- Multidiscipline: Coordinating and combing courses from different colleges (in this cases, technology and business) is always challenging. However, good experiences have been reported. Students learn more when dealing with peers with different backgrounds;
- Environment: The course should not be limited to the classroom. Connecting with the university ecosystems (such as technology parks) and even with other stakeholders always enriches the learning experience.

To sum up, several initiatives have been put in place in order to address software startup education. However, since it is a new topic not only for the academic community, but also to the industry, there is a lot or room for further research and development.

3 Research Method

In order to study the current state of practice of software startup education in universities, we devised a qualitative, largely open-ended survey. The goal of the survey was to understand in detail how software startups are currently taught in universities world-wide. In creating the survey, papers discussing software startup courses in universities, alongside our own teaching experiences in the same area, were used to ensure that the questions covered all aspects of such courses, ranging from duration to group size where applicable. Though some questions were given multiple choice answer options, most of the survey consisted of open-ended questions. Open-ended questions were utilized to gather data as rich as possible with a survey while still consuming less resources from the responder than a qualitative interview would have. Similarly, a survey was selected as the method of data collection over interviews due to the nature of the phenomenon being studied. Though interviews would no doubt have achieved the same goal, we considered the resource-intensiveness of interviews to be a problem when interviewing other scholars. Furthermore, university education as an area of study and course-based university teaching is a well-understood phenomenon that can arguably be comprehensively covered with pre-determined questions.

The survey contained questions about both the course and the teacher(s). Aside from the way software startups are being taught, we were also interested in understanding which disciplines were concerned with them the most. In addition to focusing on teaching methods, the questions also covered the basic course information: course length, course name, which discipline the course is a part of, whether the course is mandatory or optional and other such generic university course information. Aside from asking how the course is held, we also aimed to find out some of the reasoning behind the choices by asking some why-based questions. The survey in its entirety can be found in the following link: *https://bepidpoa.typeform.com/to/kuh8bK*.

The survey was sent out to individuals involved in teaching software startups in universities. Aside from contacting such individuals we knew beforehand, we searched for such courses online and contacted the teachers. However, this survey is still on-going, and we are in search of more responses from those involved with teaching software startups, as we will discuss further in the following sections.

4 Preliminary Results

Though the survey is still on-going, and we wish to gather more responses before presenting further analysis on the subject, in this section we present preliminary results based on the 10 responses gathered thus far. Perhaps due to the nature of software startups, all of the responses so far have described courses either involving a high degree of practical work or focusing entirely on practical project work on a hypothetical or real software startup. As software startups operate under a lack of resources, have little to no operating history, and typically consist of inexperienced (developers or otherwise) individuals [22], it is indeed possible and even rather simple to replicate or simulate experience in a university course setting, just as it is possible to have the students attempt to found a real-world software startup in the process. Indeed, all courses were described to be practice-oriented courses involving teamwork.

In relation to our second research question (RQ2), software startups are software companies operating in terms of academic disciplines, in an area combining business and information technology. This was also reflected in the responses. Eight of the ten courses were open to either a combination of IT and business students, or all students regardless of their major. Furthermore, all of the courses described in the surveys involved team-based work between students, and largely encouraged multidisciplinary teams including both business and IT students, as well as others if applicable. Student team sizes in the courses were varied but the common consensus was that at least three students would ideally be in a team as “2 is not a team, it is a pair”, as one of the responses remarked. Conversely, five students per team was generally considered to be a soft upper limit, with multiple responses arguing that more than five students in a team would be likely to create problems in work distribution among the team.

Whereas all of the courses involved practical work, the nature of it was varied between responses. Some courses were more focused on software engineering with a secondary focus on the entrepreneurship aspect, whereas other courses were more focused on the entrepreneurship and innovation aspect with a secondary, if any, focus on practical software engineering. In two cases, the student teams would work on external commissions from real-world customers, although the trend seemed to be that the students were expected to develop their own ideas. These ideas, then, were worked on during the courses, and while they were never required to become real software startups, the students were typically encouraged to do so. In some cases, the students had indeed gone on to create successful real-world startups based on their ideas from the courses.

A clear line between a mock-up startup and a real startup in the courses described in the responses was not generally drawn. Even though the startups were not all intended to be real-world startups, or to become ones at a later point in time, all teams were expected to validate their ideas in some way, verifying that they would satisfy a real need. This typically meant carrying out surveys and interviewing potential customers, or even creating actual landing pages and social media profiles for the course startup. This was also the approach used for other work on the startups: for instance, in one of the courses everyone would pitch to real investors at a course end event, even if they had no plans of actually continuing to work on the idea after the course. In this fashion, teams that wanted to create a real startup based on the idea were free to do so without needing to take any actual steps, and the ones that were there purely for educational purposes nonetheless created a mock-up startup as if they had been working on a real one. Only one of the courses was described to be purely educational.

Past these similarities, however, the way the courses were carried out on the level of smaller details was highly varied. For example, in terms of deliverables or gradable tasks, some courses would require the students develop a working piece of software whereas other courses would focus more on honing the idea and then pitching the idea as the final result of the course. In the cases where software development was to be carried out, agile methods, mostly *ScrumBut*, were typically followed, but on the other hand programming language and platform were typically not pre-determined. Seeing as the idea being carried out largely determines how it could (or should) be done, this is understandable unless the course is more focused on teaching, for instance, mobile application programming for Android while simultaneously teaching startup entrepreneurship. The way the students were supervised during the course also highly depended on the required deliverables of the course.

Though the courses focused on practical work, they featured weekly or otherwise regular lectures. Aside from teaching relevant theories such as the Lean Startup Methodology [19], the lectures were typically used to support the practical work more closely as well. Past the lean startup methodology, little consensus existed on which methods or theories to teach. In fact, the learning goals for the courses were notably varied, which serves to highlight the differences in the foci of the courses. Learning goals listed in the responses included:

- Strategies to test out business hypotheses;
- Practical programming skills;
- Project management skills;
- Helping students discover which aspects of entrepreneurship they like the most personally;
- Innovative business practices;
- Being a startup practitioner;
- Agile software development methods;

- Team skills;
- Using practitioner tools such as GitHub;
- Entrepreneurship.

Based on the number of responses so far, we have outlined some of the general trends in the way software startups are taught in universities. However, this research is still on-going, and based on our current set of data we are as of yet unable to provide conclusive answers to our research questions. The more general trends in the way software startups are taught can already be seen in the data in order to provide a tentative answer to our, but our sample size is still too low to go into further detail.

5 Final Remarks

To summarize our findings, regarding que first research questions - *Aside from conventional lecture-based courses ending in an exam, how are software startups taught in universities?* - the courses focus on carrying out practical work, either in the form of software engineering, creating a startup idea and developing it further, or both. The courses generally involve creating a mock-up startup in student teams and, at minimum, coming up with an idea and developing it into a business plan. No clear line is usually drawn between mock-up startups and real startups in that the student teams are expected to carry out the same tasks regardless of their own goal with their course startup or startup idea. The courses often encourage students to create a real startup with their idea but do not require them to do so.

In regard to the second research question - *How do these courses deal with the multidisciplinary nature of a software startup (business, technology, design)?* - some of the courses focus primarily on one aspect of software startups such as software engineering and practical programming. These courses are typically only open to students of that subject such as software engineering. However, most courses seem to either involve students from different disciplines, typically from business and IT ones, in order to create multidisciplinary teams. Such multidisciplinary courses seem to be more common than those focused aimed at only students of software engineering, for instance.

Another important point is that startup-related concepts are seen as an integral part of entrepreneurship by now. Notably, the courses were not necessarily referred to as startup courses of any kind. In fact, only three out of the ten responses so far had the construct *startup* in the course title. The course titles were more associated with innovation, entrepreneurship, and software engineering practice.

Finally, it is very likely that courses described in academic papers present non-conventional educational ideas rather than tried-and-true methods for teaching. In our opinion, there is no reason to write a paper about a lecture-based course on software startups that ends in an exam about a book and the course content. Thus, in contacting the authors of various papers in relation to our survey, the data has become biased in this fashion. It is unlikely that all or even most courses on software startups would be so focused on practice, even though it would appear that the amount of practice-focused courses in the area could be higher than usual.

References

1. Blank, S.: The Four Steps to the Epiphany: Successful Strategies for Products That Win. K&S Ranch, Incorporated (2013)
2. Blank, S., Dorf, B.: The Startup Owner's Manual: The Step-by-step Guide for Building a Great Company. K&S Ranch, Incorporated (2012)
3. Buffardi, K., Robb, C., Rahn, D.: Tech startups: realistic software engineering projects with interdisciplinary collaboration. Journal of Computing Sciences in Colleges **32**(4), 93–98 (2017)
4. 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. pp. 9–12. CHASE '18, ACM, New York, NY, USA (2018)
5. Chanin, R., Sales, A., Pompermaier, L.B., Prikladnicki, R.: A systematic mapping study on software startups education. In: EASE. pp. 163–168 (2018)
6. Coleman, G.: An empirical study of software process in practice. In: System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on. pp. 315c–315c. IEEE (2005)
7. Crowne, M.: Why software product startups fail and what to do about it. Evolution of software product development in startup companies. IEEE International Engineering Management Conference **1**, 338–343 (2002). https://doi.org/10.1109/IEMC.2002.1038454
8. 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
9. 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). pp. 332–338. IEEE Computer Society, Las Vegas, NV, USA (July 2008)
10. Génova, G., González, M.: Educational Encounters of the Third Kind. Science and Engineering Ethics **1**, 1–10 (2016)
11. Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software Development in Startup Companies: The Greenfield Startup Model. IEEE Transactions on Software Engineering **42**(6), 585–604 (2016)
12. Harms, R.: Self-regulated learning, team learning and project performance in entrepreneurship education: Learning in a lean startup environment. Technological Forecasting and Social Change **100**, 21–28 (2015). https://doi.org/10.1016/j.techfore.2015.02.007
13. Hilmola, O.P., Helo, P., Ojala, L.: The value of product development lead time in software startup. System Dynamics Review **19**(1), 75–82 (2003)
14. Kon, F., Cukier, D., Melo, C., Hazzan, O., Yuklea, H.: A panorama of the israeli software startup ecosystem. Available at SSRN 2441157 (2014)

15. Kontio, J., Ahokas, M., Poyry, P., Warsta, J., Makela, M., Tyrvainen, P.: Software Business Education for Software Engineers: Towards an Integrated Curriculum. 19th Conference on Software Engineering Education and Training Workshops (CSEETW'06) pp. 4–7 (2006). https://doi.org/10.1109/CSEETW.2006.15
16. Osterwalder, A., Pigneur, Y.: Business model generation: a handbook for visionaries, game changers, and challengers. John Wiley & Sons (2010)
17. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software Development in Startup Companies: A Systematic Mapping Study. Information and Software Technology **56**(10), 1200–1218 (2014)
18. 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 (2015)
19. Ries, E.: The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses. Crown Business (2011)
20. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business (2011)
21. Sutton, S.M.: The role of process in software start-up. IEEE Software **17**(4), 33–39 (Jul 2000). https://doi.org/10.1109/52.854066
22. Unterkalmsteiner, M., Abrahamsson, P., Wang, X., Nguyen-Duc, A., Shah, S., Bajwa, S.S., Baltes, G.H., Conboy, K., Cullina, E., Dennehy, D., et al.: Software startups–a research agenda. e-Informatica Software Engineering Journal **10**(1) (2016)

Effectuation as a frame for networking decisions – the case of a Finnish information technology start-up

Katariina Yrjönkoski and Anu Suominen

Tampere University of Technology, Pori, Finland

Abstract. Effectuation is an emerging theory framework for managerial decision-making, particularly in the context of start-ups. It attempts to capture the nature of managerial decision-making in circumstances where the essential business artefacts, like markets, customers or business model, do not exist yet or are undergoing a revolutionary transformation. This study focuses on applying the effectuation theory to inter-organizational relationship building in the early phases of network formation. According to the theory, effectual processes are characterized by four main principles: 1) a focus on projects where the loss, in worst-case-scenario, is affordable 2) a focus in a short-term experiment to identify opportunities 3) control over an unpredictable future 4) a focus on cooperation to control the future. In this article, the subject is approached by considering those four principles and capturing the acts that manifest them in practice. The findings show that all four principles of effectuation are detectable in the early relationship building. The case company uses effectual processes to balance the uncertainty of the environment and to capture the arising opportunities. Furthermore, the case results suggest a tendency to adopt more systematic processes towards collaborators, as the business transforms into more stable one. The results complement previous research by illustrating the manifestations of four principles of effectuation in inter-organizational acts in practice. Further research should be devoted to revealing the diverse and versatile multidimensional nature effectual and causal models toward collaborator selection and processes that manifest them, instead of the current dichotomy of predictive and non-predictive strategies toward collaborators.

Keywords: Effectuation, Network, Relationship building, Start-up, Non-predictive strategy

1 Introduction

The mainstream of strategic and managerial literature consists of prediction and planning based frameworks (e.g. [1, 21]). However, while they are widely approved and utilized by thousands of successful business managers, they may not be useful in uncertain environments, such as start-up companies. Effectuation is an emerging theory framework for managerial decision-making. Effectuation aims to explain the behavior that is typical in uncertain environments of companies and to highlight a certain type of benefits that might be attained particularly by effectual processes [5, 9, 23, 25, 26, 29]. Therefore, the objective of this paper is to increase understanding of the trajectory of the start-up companies regarding their relationship building. Conforming to OECD's view, [20] we consider start-up as an early phase of an enterprise. We focus on the acts of companies in their early phases of networking, which may manifest the effectuation in their relationship building attempts. This paper discusses the four principles of effectuation presented in the previous literature that start-up companies may use in order to balance the dilemmas of their uncertain environments, which is inherent in emerging business areas and start-up companies. By responding to the call by Sarasvathy [25] for further research of manifestation of effectuation in a start-up context in practice, our empirical case study is guided by two research questions:

> *RQ1: How the effectuation principles manifest as inter-organizational acts of networking in practice in a start-up company?*
>
> *RQ2: How the shift between the effectuation and causal networking process manifests in a start-up company in practice?*

These questions deal with the effectuation principles that manifest when inter-organizational relationship and network building is a crucial element of start-up success and survival, such as in information technology (IT) industry. By studying the manifestations of effectuation principles in practice, we aim at seeking further, especially practical, insight to support management's decision-making, particularly in start-up companies. Our current belief is, that these questions on balancing the uncertain environment of start-ups by exploiting the available resources and controlling them, instead of predicting, are pondering many firms and their management, which the previous research of traditional, predictive strategies concerning relationship building has not been able to tackle. Therefore, we approach the subject by categorizing the discovered the manifestation of the effectuation theory according to the four principles of effectuation. The paper presents a single case of an egocentric network during the pre-networking and network formation phase in IT-industry. The goal of the paper is to identify and present the effectuation behavior the focal company uses during its early phase relationship building. Although the current literature of effectuation relates closely to both strategic and entrepreneurial research, in this study the topic is considered at the organizational level, as organizational, not an entrepreneur's individual behavior. In her seminal work, Sarasvathy defines effectuation and causation as "processes" and the main structures of effectuation as "principles". However, depending on a context, some other terms are used as well: effectuation and causation are considered as "approaches", "behavior" or "decision-making logic", to mention some. For clarity matters, we hold to terms "process" and "act", in a meaning of the manifestation of effectuation, regardless whether they are higher-level networking policies or more practical operations.

This study proceeds as follows: first, we suggest that effectuation, as a phenomenon concerning inter-organizational relationship building, is a preliminary phase in a continuum of more systematic and predictive approaches to networking, and is dominant until the business is properly established. Second, we enrich the general theory of effectuation with a single-case study in which we identify preliminary networking acts that portray the four principles of effectuation theory in networks. The examined case includes top and middle-level managers describing their early relationship building processes in IT-industry.

The paper begins by exploring the literature related to the inter-organizational relationships and networks, and the concepts of effectuation. The relationship between effectuation and networking is also explored briefly. We then introduce the qualitative research design, including a single-case study of a start-up company operating in the IT industry. The findings section presents the manifestation of effectuation of the case company, viewed from the four principles of effectuation.

2 Effectuation in inter-organizational networks

2.1 Inter-organizational relationships and networks

Since the 1970's researchers been interested in networks and considered them as a strategic response to dynamic environmental pressures, thus naming it the network paradigm[7]. The network paradigm has yielded a vast literature of inter-organizational entities (IOE)[8], e.g. alliances, coalitions, clusters, partnerships, strategic alliances, strategic business 'nets', and networks. These IOEs have inter-organizational relationships (IORs) with each other and engage in inter-organizational acts, such as working together, collaborating, networking, contracting, outsourcing, cooperating, partnering etc. [8] Yet *"no single grand theory of networks exists"*[22]. As the majority of IORs fail, it has been suggested that the management practices and techniques that facilitate the ongoing success of IORs should be researched[3]. For example, there is very little research on early phase alliance formation, the phase where little trust is detected[2].

In general, networks are viewed in two main ways with two underlying assumptions about networks and their management: either that they are unmanageable emergent structures or with strategic orientation[14]. Although, the view of strategic orientation, *"networks are managed all the time"*, would suggest overall well-planned inter-organizational acts, it does recognize the serendipity as an inherent part of network management, that "*consists of a complex pattern of activities—intentional or emerging, strategic or non-strategic—for framing, activating, mobilizing, and synthesizing*"[16]. Therefore, our study is based on the strategic view of networks. However, we see that emergence and serendipity are part of a continuum of the inter-organizational acts, that on one end are intuitive and on the other end prescriptive.

As there is an abundance of literature regarding networks, there are also a whole variety of different types of networks, which have different levels of analysis: micro - the egocentric network, and macro - the overall network structure [15]. Furthermore, there are also dyadic, triadic and global levels of network analyses, and even multi-level [6]. Yet, all three levels of network analysis: dyadic, triadic and global, may take place in networks of individuals, or of units, or of organizations [19]. Furthermore, regarding the methodological issues and choices, network research can be classified into four major categories: social capital research, network development research due to the direction of causality, and two additional classes to reflect the level of analysis of interpersonal and inter-organizational level research. The network development research scholars focus on recognizing the patterns and determinants of network formation and change [4]. Network formation has been researched to some extent and there have been discovered three main phases: 1) pre-networking or network formation phase, 2) network development phase and 3) solid networking [11]. In pre-networking and network formation phases, the preconditions are established, potential partners identified and joint interests considered.

In our study, we focus to an egocentric network, i.e. a network around one focal actor. As our interest is on the network development of one focal company, we are interested in the initiation and setting up the IORs and the inter-organizational acts during the pre-networking and network formation phase.

2.2 Effectuation theory in networks

Traditional management frameworks for decision-making are strongly based on causal reasoning. The classics of managerial literature, like Ansoff [1] and Porter [21], emphasize the importance or systematic analysis and a discipline for carefully predicting the business environment. That usually includes actions like carrying out market and competitor analyses, identifying customer segments, setting a specific goal and defining a well-considered strategy to reach it. However, the impact of causal planning tools may remain unattainable if the main business artefacts, like markets, customers or business model are not established enough. Such circumstances may occur, for instance, in the start-up phase or in industry or company transformations that significantly change the prevailing business regularities. In effectuation literature, situations such as described above are usually bundled to a concept of "uncertainty". [9, 25]

In her seminal article, Sarasvathy [25] proposes that decision-makers tend to act effectually if they believe they are dealing with relatively unpredictable phenomena. This tendency appears in the dominance of experimental and means-driven acts and iterative learning techniques. [9, 25] While this kind of behavior is often considered as unfavorable deviations in causal frameworks, effectuation theory aims to offer an idea of an alternative and equally acceptable process, which may suit better for cases of high uncertainty.

Effectuation theory consists of four principles that deal with exploiting the available resources and controlling the environment instead of predicting it. Principles are general constructs that group the features of operative and decisional acts. Those four principles are listed on the following [5, 25]:

1. The Principle of Affordable Loss. Decisions are driven by the idea of affordable loss, rather than optimizing profit. The organization focuses on projects where the loss in a worst-case scenario is still affordable.
2. The Principle of Partnering. Decisions are characterized by active partnering attempts, rather than conducting competitive analyses. The organization tends to collaborate to control the future, as they can not predict it.
3. The Principle of Exploiting the Contingencies. Decisions focus on exploiting the contingencies to remain flexible, rather than exploiting the pre-existing knowledge.
4. The Principle of Control. Decisions and actions aim to control the future rather than predict it. Short-term experiments are preferred to identify business opportunities.

Effectuation and causation are often described as a dichotomy or two extremes of a continuum. Causation is consistent with rational decision-making perspectives, which is mostly based on analyzing, predicting and planning in such conditions where the environment and outcomes are predictable. While effectual processes take particular means as given and focus on selecting between possible effects that can be created with them, causation processes take a particular effect as given and focus on selecting between means to create the effect. [5, 25] However, previous literature suggest that in reality, causation and effectuation may occur simultaneously, overlapping or intertwining over different contexts of decisions. Yet, it has been proved that a path to new venture may follow a well-defined causal path but as well a well-defined effectual path. An important venue for further research is to determine the circumstances under which each approach is more appropriate. [5, 25]

Network orientation is an integral part of effectuation. When looked through effectual lenses, all business is a matter of an effectually born network exploiting the opportunities and converting them into new artefacts, i.e. new business, solutions or even new markets. The effectual network is a dynamic construct of interactions between stakeholders. The network is initiated by pre-commitments between the first set of actors and grows iteratively over time. Every new stakeholder committing the network extends the resources available for the network - but also sets new constraints to it. While the network grows and the outcome, i.e. business, market, etc., turns to more stable and predictable, the network tends to turn less effectual, too. [26]

Internationalization, as a specific case of networking [17], has many characteristics of effectuation. Instead of being a formal search and selection process, the foreign market selection seems to emerge from opportunities presented by network members. Moreover, especially when conducted under uncertainty, internationalization attempts in small firms are intentionally carried out in an effectual manner. [12, 13]

3 Research process: method and case description

We chose a case study approach [10, 27, 28] to empirically search the manifestations of the principles of effectuation theory. First, we examined the literature on network theory, as well as the theory of effectuation, particularly in a network context. Then, we selected one start-up company together with its egocentric network as the case study. This particular company (described below) was chosen because of its idiosyncrasy: it is a start-up company, yet stable enough to reflect their path of early phases of networking. Thus, the case is interesting from both academic and practical viewpoints and gives unique information about the early phase acts regarding relationship building.

COMPANY LTD (name anonymized due to confidentiality reasons) is a Finnish IT startup firm. It was established in 2017 as a spin-off of another software company. Its core business is to develop a cloud platform for digitalizing certain public administration processes, which are typically participated by companies and local authorities. The service is based on open source; the source code is available in GitHub platform. Currently, COMPANY LTD's network consists of external information system and data service operators, as well as local authorities, BtoB, and BtoC customers. The company has also progressed in its internationalization efforts lately and concluded an agreement of their first international service implementation.

The research material was composed of theme interviews that were supported by network pictures drawn by the interviewees. The manifestations of the effectuation principles regarding relationship building in the early phases of the case start-up company was studied by interviewing its four members of senior management. The interviewees were also asked to draw a sketch of their network. Interviews were carried out as theme interviews. As they were intended to give preliminary understanding whether the effectuation principles are detectable, the terms of effectuation were not used in the questions and interviewees were allowed to describe their networking acts rather freely. Another reason for forming the questions very neutral was the aim to avoid social desirability bias. As causal logic is a strong norm in managerial discipline, highlighting the effectual behavior could have led the interviewees to answer in a manner that they supposed to be viewed favorably. Interviews were transcribed to text files, which were then carefully analyzed. The analysis of the research material was carried out with Atlas.ti. The qualitative interview material was coded mainly with structural coding. Based on the theoretical setting for this study, we used pre-defined categories and codes for the four main principles of effectuation. Structural coding is appropriate to label conceptual phrases that relate to a specific research question used to frame the interview. It both codes and categorizes the data. Further, in vivo coding was also utilized. In this study, structural codes were used to label the phrases indicating each of the four effectuation principles. In vivo coding was utilized to label other effectuation related themes, which were identified as potentially meaningful concerning our research questions. [24] In practice, the phrases indicating each of the principles were identified. In addition to identifying the main effectual main principles, all the phrases referencing to shifts between effectuation and causation were also gathered. They were used to analyze the factors the interviewees relate to the transition between these two logics.

4 Results

Analyzing the material resulted findings on both research questions: the manifestation of effectuation principles and the transition between effectual and causal logics. In addition, some significant findings on overall effectual network approach arose. In this chapter, we will present the findings divided in mentioned three categories in the following order: first, we report the findings on overall effectual network approach, second, we describe the findings on the manifestations of the effectuation principles, and last, we highlight some interesting points related to transition between effectuation and causation.

4.1 Findings on overall effectual approach

In our analysis of the interviews, we discovered that the top management and middle management differed by their focus relating networking. While top managers were more likely to describe strategic decision-making, high-level principles and approaches to networking, middle management dealt mainly with operational networking themes, like knowledge and collaborative issues. This may be a logical reflection of their daily work assignments. However, this resulted in the top managements' answers being more informative when looking for understanding the effectual and causal approaches behind the acts.

In general, interviewees seemed to use a strongly effectual framing when considering networks. They mentioned the network as a crucial factor in creating the new solution - or, a new ecosystem, as the case company wants to do. The phrases they used to describe the role of the network indicated the belief of the network as a driving force for creating new business. This reflects the idea of the effectual network as an enabler for the transformation process, i.e. the process of converting contingencies and opportunities to a new business together with network stakeholders.

In the interviews, the interviewees described their mindset towards relationship building and distribution of resources with more capable partners:

> *”But surely there is also our COMPANY LTD’S ideology, about the code being an open source code and we talk about open interfaces, operations and culture. Furthermore, in a way we want to enable that we can operate with everyone. My own thinking has been that if there is someone, who does things better than us, it makes sense that they do it and we concentrate on things that we manage to do better than anyone else. Exaggeratedly put “we do everything by ourselves”- thinking, is quite an old school to me, we rather exploit the best know-how.“*

The interviewees expressed that relationship building and distribution of resources with more capable partners is a sensible and contemporary act, as it enabled their own concentration on their core competencies and thus overall better performance. This expression can be interpreted as a manifestation of overall effectual thinking that emphasizes the exploiting of network resources and knowledge. Partnerships were seen as crucial for innovating and/or creating new business. When the interviewee was asked about discovering the "blue ocean" for strategy with the partnership, the interviewee expressed it to be a self-evident truth.

> *"Yes. Do you believe that the blue ocean is found especially in collaboration then?"*
> *"It is not a matter of faith, it is crystal clear…[laughter]…"*

This expresses an essential matter i.e. it is believed that the emerging potentialities will mould into a real business in collaboration with the network.

4.2 Findings on the effectuation principles

The main finding on the effectual main principles was the discovery of all four of them. However, there were differences on the clarity each of the principles appeared in the interviews. The principle of partnering and the principle of exploiting contingencies were easy to detect as they emerged in several expressions. The principle of affordable loss and the principle of control were expressed in a weaker manner, but still detectable. In the following, we will present the findings interviewees' networking discourse concerning each of the four main principles.

The Principle of Exploiting the Contingencies. This principle covers an idea of all acts being driven by utilizing the resources and opportunities at hand, rather than building the strategy strictly on pre-existing knowledge. The interviewees highlight in several comments their attitude on the network as a source of collaborational innovations, more than a source of pre-defined capabilities and resources. One interviewee even emphasizes the benefits that are more likely to reach when acting effectually: when the goal and plan are not too fixed, innovational ideas may be more likely to appear.

> *"Well, then could good innovations be outlimited by accident if the things are defined too precisely beforehand."*

Here it is expressed those benefits the effectuation may bring, i.e. viewing the company's environment more broadly, thus enabling discovering innovations that causal behavior might outlimit.

> *"… and then I saw very clearly, that if we try to penetrate the markets, the only choice we have is to begin to change the rules of the game, and as we are small, we have to seek the teammates to do the change with us."*

Above the interviewee expresses the network and partners as an only choice to transform the contingencies into new business.

The interviewees also describe the partnership to have a trajectory, which is in many phrases characterized by the depth of trust. The deeper the partnership is (or is aimed), the stronger is the trust needed. Experimenting with partners is described also as a mechanism for building the trust between companies.

The Principle of Partnering. The interviewees had a clear mindset of partnering as an inborn and favorable culture in their company. They considered it a contemporary and almost self-evident way of action. Further, they emphasized win-win –situation as a necessity and expressed a fair play and loyalty even when it was not the most profitable option for the company itself.

The interviewee illustrated the essential bidirectional benefits of partnering:

> *"you have to be sensible, meaning the partnership cannot work if both parties do not benefit from it"*

> *"… and when you have the win-win –thinking, you don't have to think about who has the most of power."*

The interviewees expressed that the win-win-thinking is mandatory for successful partnering and articulated the act of complying also to the needs and requirements of the counterparty, thus manifesting positive mindset towards networking. Network that benefits all parties is considered as a motivator for collaboration, which even minimizes the need for formal commitments in the current phase.

The interviewees are aware that competitive settings may emerge, but they still do not feel reasonable to focus on analyzing the competition. On the opposite, they mention it to be an old-fashioned manner.

> *"I present this in a bit pointed way now, but in my opinion, it is quite old-fashioned thinking." (speaks about using many resources on analyzing competitors)*

> *"Of course challenges may raise, we may start developing the service, and then we step on the feet of a certain partner. Then it has to be discussed, whether we can still work together or will it cause a break between us. "*

The Principle of Affordable Loss. This principle had the weakest reflection in this dataset. That may be due to its nature: whether to be driven by managing the risk or by optimizing the profit may be such an unconscious and personal behavior, that identifying it in verbal expressions may require questions that are more specific. However, the interviewees described their networking overall as very open and trusting, though they still are aware of potential competitive settings and other risks.

> *"… ok, we do not want to be too naive, we tell a lot to the external world and to partners, but not everything. But mostly, if we have some plans, we also tell them. The plans may not always come true, but so what?"*

The interviewees expressed in several phrases their win-win-attitude and sustainability in resource utilization. They all agreed that it is most reasonable for everyone to focus on own core business and to acquire the rest from the network – even when it will not optimize their own profit.

The Principle of Control. This principle reflects the tendency to control the future, as well as they, can, instead of trying to predict the future which is considered too difficult in the current phase. The interviewees describe the difficulty of predicting, and mention they rather experiment with different partners. Experiments and pilot projects “separate the sheep from the goats”; if there is no trust and common business interests, after all, the partnership dies or turns to standby.

> *“... well, then it is such, that it is very seldom possible to know beforehand is it going to work or not. “*

> *”We are still so young company, so much is happening all the time. So it is just the best way to conduct a pilot to see if it is going to work. “*

Although the interviewees mention that plans and high-level strategies also exist, they still seem to prefer experimenting as the best way to find the good partner matches in the current phase.

4.3 Findings on the transition between effectuation and causation

The data showed us the dynamic nature of both effectual and causal networking logics. Although effectuation seemed to be the dominating behavior at the moment, all of the interviewees indicated an idea of moving towards more causal and systematic practices on partner selection. When this would happen, seemed to be imagined as some sort of maturity level or turning point. The interviewees expressed repeatedly that now they are acting effectually and felt it acceptable, but more systematic would be needed, and was already under development.

When the interviewee was questioned, whether there is any systematism in the networking acts, he responded:

> *“No, so far nothing can be described as systematic. These municipal system suppliers have come to us as given. Regarding the service development side, it has been like pulling ropes that have come to our way. However, now we have also a clear aim, which we have to take care so that this balance stays, and a kind of prioritizing of the partnerships that are really [important].*

This is an expression of current, unsystematic effectuation logic, that from now on with the aim of “life cycle”, will also grow to more systematic, even causal encompassing networking acts.

> *”well, I have considered it good, that it has not been too formal with us, but we have been able to react fast and gone with the flow according to the situation, and in a way, it has been ok to remain a certain kind of intuition. However, now that the volumes are growing and personnel increasing and operations expanding, we are forced to include some kind of formality. “*

Effectuation is seen even beneficial in the early phases of business development, however as portrayed, there is an idea that in the long run, more formal i.e. causal handling of business would serve better. This could also be interpreted that the organizations might perceive that there exists a maturity point, after which more systematic planning could be feasible.

The interview data gives a notion of a turning or maturity point, in which some set of elements will be known and stable enough to handle with causal planning tools.

> *"…but we have passed that phase, where we can carry on only by a sort of gut feeling."*

This expresses identification of a sort of breaking point after which it would be clearer or more beneficial to plan with causal logic. At the same time with carrying out the interviews, we allowed the participants also to draw a sketch of their current network, if they felt so. Some of the drawings reflected a systematic in partner selection; current partners were categorized by the type of their output services or products. Partners were also divided into "collaborative partners" in a meaning of common marketing and revenue sharing, and to "suppliers" in a meaning of the mostly unidirectional buyer-seller relationship.

> *"... or, it is so that, if it is in disorder in the beginning, so we should try to build more systematics, but there will always be some part that is still in disorder. And what is the balance between things in order and things in disorder, it is varying all the time and it is ok."*

Here the interviewee indicates multidirectional fluctuation between effectuation and causation. When asked about partner-related risks, i.e. ownership and other juridical questions, the interviewees expressed a need for more formal commitments in the future. Especially the data owner questions may require more careful consideration and agreements.

> *"So far it has been relatively clear, but it can be that it is an area that will become more challenging in the future. And even now there is that – yet there have not been such situations - but now that a lot of material is accumulated to our service and if, for example, [customer X] will give up at some point, who has the ownership to the material – well [customer X] of course, because we are like an Outlook, we enable the data transfer. But this is not necessarily clear to everybody, and now that we have these further plans, for example, one of the focal issues in the near future that is talked about is the selling the data, data selling."*

While this reflects the effectual way of establishing pre-commitments, it is also a sign of emerging need for more causal partnering process.

5 Discussion

Understanding the processes the companies adopt for their relationship building in the early phases of their development, and consequently, the effects of the made acts have on the success of the business is one of the most significant fields of research to help companies to success and develop competitive advantage in current, fragmented business environments. The importance of networking as an integral and strategic part of any business has been recognized for decades. However, the frameworks for networking strategies have been mainly tied to a causal management norm, which has left the managers to lack tools for tackling early phase networking dilemmas. In her seminal work, Sarasvathy shed light on a new insight of considering the acts of uncertain phases as a valuable logic with idiosyncratic benefits. Further exploration of this approach would lay the ground for developing systematics and strategic frames for these phases of a business trajectory, which, moreover, could even boost the innovational activity as the effectual processes allow the companies to experiment with more numbers of new ideas with lower costs. As a Finnish philosopher Frank Martela states: “The less prepared ideas often capture the highest innovation potential.”[18]

Besides causality, effectuation is regarded as one method of decision-making attitudes for relationship building in companies’ networking endeavors. However, there is a lack of empirical cases that illustrate the usage of effectuation logic in companies’ decision-making interrelated with their relationship building. Therefore, this article responds to the research need and contributes by portraying a case of one start-up company with its egocentric network operating in information technology. The article contributes to the theory of effectuation in network context, and more specifically to the concept of effectuation by highlighting the manifestations of four effectuation principles in practice. Our results showed, that all the four effectuation principles earlier introduced in the literature conceptually, do manifest as inter-organizational acts in the case company’s relationship building in the early phases of its formation. Thus, this article confirms the previously conceptual views of the effectuation of Sarasvathy and her colleagues, as well as Chandler [5, 9, 23, 25, 26, 29] as a valid decision-making logic for companies in an uncertain environment and/or situation, such as in the start-up phase. Moreover, our results showed that effectuation as a phenomenon is more detectable in the discourse of the top managements, thus implying that effectuation processes are, in fact, an integral part of strategic goal-setting and decision-making, as suggested by Sarasvathy and Chandler, than of daily operations of putting the chosen strategy into practice.

All of the four principles of effectuation theory, i.e. The Principle of Exploiting the Contingencies, The Principle of Partnering, The Principle of Affordable Loss, and Principle of Control, manifested in the case results. The Principle of Exploiting the Contingencies manifested as acts being driven by utilizing the resources and opportunities at hand. Thus making the novel innovations and business opportunities prospective, yet depending on the situation, requiring a trajectory of trust. The Principle of Partnering manifested as a culture of win-win based collaboration, that is viewed contemporary and almost self-evident way of action. The Principle of Affordable Loss, manifested as the focus on own core business even if it does not optimize the company’s own profit.

The Principle of Control manifested as a preference to experiment regarding potential partners in order to control the future, rather than pursuing to predict it. However, there was a difference in the clarity, those four principles of effectuation manifested in the case organization's inter-organizational acts. In our case, The Principle of Affordable Loss manifested with the weakest reflections, which may be due to its nature. The Principle of Affordable Loss incorporates behavior of either risk prone or risk avert, thus being dependent on the personal traits of an individual. However, it might also give different results at inter-organizational level depending on the business and industry the organization is operating in. However, our results indicate that the four principles of effectuation may not be equally represented in a single company, but manifest more or less evidently depending on the trajectory the company is in. However, as there is a lack of cases of manifestations of effectuation in practice, further cases would shed more light to the potentially different levels of existence of the four principles of effectuation in the various phases of the company's development. Moreover, since there is an abundance of literature regarding effectual behavior of an individual entrepreneur, and lack of the descriptions of the phenomenon of organizational behavior, the effectuation theory would also benefit from the study both at individual and organization levels.

Additionally, our results showed, that the effectual behavior offers the companies a chance to "play the field", seek relationships and opportunities that may lead to successful business endeavors or innovation, that potentially would not have been detected with traditional causal decision-making behavior/logic. Particularly in the early phase of the company, as well as in the early phases of the networking the effectual behavior is considered having many positive benefits. When regarding networks with effectual lenses, they are a fundamental organism for any new venture creation. This finding contributes to the networking literature by accompanying the previously recognized need for a wider spectrum of strategic collaboration building frameworks [e.g. 22].

Furthermore, our findings imply that while the dilemmas of uncertain environments of start-ups are dealt with effectual processes, they tend to turn to more predictive in time. Results also imply that the shift from effectuation-oriented logic to causal-oriented one may require a certain maturity or turning point to be reached. As we detected several indicators of this kind of turning point or zone, we consider this as a significant venue for further research. Exploring the attributes of this maturity point would give valuable understanding of multidirectional and intertwining nature of effectuation and causation. As a contribution, we suggest that instead of dichotomous, adversarial discussion of either non-predictive or predictive strategy of relationship building, a strategy spectrum, which depending on the company's maturity may have more or less characteristics of effectuation, even concurrently depending on the partnership/relationship at hand, is acknowledged. As the effectuation theory in inter-organizational settings is still in its infancy, in general, clarification of the concepts would benefit the theory development.

As a practical implication, this study clarifies the effectuation behaviour as a potential and functional approach to managers for relationship building in the early or otherwise uncertain phase of the company's development.

As a single case study, this research does have its limitations. Particularly, as being a single case study it portrays only a specific view on effectuation. However, as being a carefully selected case, this study gives valuable preunderstanding of effectuation as networking-related phenomena. After all, despite its limitations, this study succeed in enriching the verification of the rudimentary theory of effectuation and in pointing some interesting avenues for further exploration.

References

1. Ansoff, H.I.: Strategic Management. Macmillan, London (1979).
2. Badir, Y.F., O'Connor, G.C.: The Formation of Tie Strength in a Strategic Alliance's First New Product Development Project: The Influence of Project and Partners' Characteristics. J. Prod. Innov. Manag. 32, 1, 154–169 (2015).
3. Barringer, B.R., Harrison, J.S.: Walking a Tightrope: Creating Value Through Interorganizational Relationships. J. Manage. 26, 3, 367–403 (2000).
4. Carpenter, M.A. et al.: Social Network Research in Organizational Contexts: A Systematic Review of Methodological Issues and Choices. J. Manage. 38, 4, 1328–1361 (2012).
5. Chandler, G.N. et al.: Causation and effectuation processes: A validation study. J. Bus. Ventur. 26, 3, 375–390 (2011).
6. Contractor, N.S. et al.: Testing Multitheoretical, Multilevel Hypotheses about Organizational Networks: an Analytic Framework and Empirical Example. Acad. Manag. Rev. 31, 3, 681–703 (2006).
7. Cravens, D.W. et al.: New Organizational Forms for Competing in Highly Dynamic Environments: the Network Paradigm. Br. J. Manag. 7, 3, 203–218 (1996).
8. Cropper, S. et al.: The Oxford Handbook of Inter-Organizational Relations. Oxford University Press, Oxford (2009).
9. Dew, N. et al.: Effectual versus predictive logics in entrepreneurial decision-making: Differences between experts and novices. J. Bus. Ventur. 24, 4, 287–309 (2009).
10. Dyer, W.G., Wilkins, A.L.: Better Stories, Not Better Constructs, To Generate Better Theory: A Rejoinder To Eisenhardt. Acad. Manag. Rev. 16, 3, 613–619 (1991).
11. Ebers, M.: Explaining Inter-Organizational Network Formation. In: Ebers, M. (ed.) The formation of inter-organizational networks. pp. 3–40 Oxford University Press, Oxford (1997).
12. Evers, N., O'Gorman, C.: Improvised internationalization in new ventures: The role of prior knowledge and networks. Entrep. Reg. Dev. 23, 7–8, 549–574 (2011).
13. Galkina, T., Chetty, S.: Effectuation and Networking of Internationalizing SMEs. Manag. Int. Rev. 55, 5, 647–676 (2015).
14. Huuskonen, A., Kourula, A.: A Contingency Model of Network Management – Consolidating an Emerging Cross-disciplinary Field. In: 28th Industrial Marketing and Purchasing (IMP) Annual Conference, Rome, Italy, 13.-15.9.2012. pp. 1–32 , Rome (2012).
15. Ibarra, H. et al.: Zooming In and Out: Connecting Individuals and Collectivities at the Frontiers of Organizational Network Research. Organ. Sci. 16, 4, 359–371 (2005).

16. Järvensivu, T., Möller, K.: Metatheory of network management: A contingency perspective. Ind. Mark. Manag. 38, 6, 654–661 (2009).
17. Johanson, J., Vahlne, J.E.: The Uppsala internationalization process model revisited: From liability of foreignness to liability of outsidership. J. Int. Bus. Stud. 40, 9, 1411–1431 (2009).
18. Martela, F.: Hyvä johtoryhmätyöskentely edellyttää kolmea asiaa, https://filosofianakatemia.fi/blogi/ketterasti-toimivan-johtoryhman-luominen-vaatii-onnistumista-kolmella-ryhmadynamiikan-dimensiolla.
19. Moliterno, T.P., Mahony, D.M.: Network Theory of Organization: A Multilevel Approach. J. Manage. 37, July 2010, 443–467 (2011).
20. OECD: Eurostat-OECD Manual on Business Demography Statistics. (2007).
21. Porter, M.E.: Competitive Strategy. (1980).
22. Provan, K.G. et al.: Interorganizational Networks at the Network Level: A Review of the Empirical Literature on Whole Networks. J. Manage. 33, 3, 479–516 (2007).
23. Read, S. et al.: Marketing Under Uncertainty: The Logic of an Effectual Approach. J. Mark. 73, 3, 1–18 (2009).
24. Saldana, J.: The Coding Manual for Qualitative Researchers. SAGE Publications Ltd., London (2009).
25. Sarasvathy, S.D.: Causation and effectuation: Toward a theoretical shift from economic inevitability to entrepreneurial contingency. Acad. Manag. Rev. 26, 2, 243–263 (2001).
26. Sarasvathy, S.D., Dew, N.: New market creation through transformation. J. Evol. Econ. 15, 5, 533–565 (2005).
27. Siggelkow, N.: Persuation with Case Studies. Acad. Manag. J. 50, 1, 20–24 (2007).
28. Weick, K.E.: The generative properties of richness. Acad. Manag. J. 50, 1, 14–19 (2007).
29. Wiltbank, R. et al.: What to do next? The Case For Non-Predictive Strategy. Strateg. Manag. J. 27, 981–998 (2006).

Teaching Lean Startup Principles: An Empirical Study on Assumption Prioritization

Matthias Gutbrod and Jürgen Münch

Faculty of Informatics, Reutlingen University, Reutlingen, Germany
[matthias.gutbrod|juergen.muench@reutlingen-university.de]

Abstract. Creating new business models, products or services is challenging in fast-changing unpredictable environments. Often, product teams need to make many assumptions (e.g., assumptions about future demands) that might not be true. These assumptions impose risks to the success and these risks need to be mitigated early. One of the principles of the Lean Startup approach is to identify and prioritize the riskiest assumptions in order to validate them as early as possible. This helps to avoid wasting effort and time. In the literature there are several different methods for identifying and prioritizing the riskiest assumptions reported. However, only little research exists about the practical application of these methods in practice and how to teach them. In this paper, we present and empirically analyze a workshop format that we have developed for teaching the prioritization of Lean Startup assumptions. We aim at raising the awareness for assumption thinking among the participants and teach them through group work how to prioritize assumptions. The results of the analysis of a multitude of conducted workshops show that the applied method did lead to reasonable results and accompanying learning effects. In addition, the participants got aware of assumption thinking and liked learning in a practical way.

Keywords: Risk Prioritization, Riskiest Assumptions, Teaching, Lean Startup.

1 Introduction

Kevin Systrom had an idea for a location sharing app where users could “check-in”, called Burbn. The programming of the iPhone app took him a few months. In Burbn users could check-in with friends who are hanging around, get points and take and post pictures. The app had many features and was therefore complicated to use. The app was unsuccessful, but Kevin Systrom started together with Mike Krieger to analyze what the customers really were doing with their app. They found that the original assumption that users will use a "check-in" feature was wrong. This could have been validated before the full implementation of the feature. The two observed that users were basically only posting photos. Together, they decided to get rid of all of the app functionalities except for sharing and liking photos. They spent months creating and experimenting with prototypes in order to validate risky assumptions. In the end, they built an app called Scotch. Scotch was slow and full of bugs. Nevertheless, they doubled down on the insight that sharing photos in a frictionless way is important for users. In the next version, they focused on a super easy to use app where the users only need three clicks to upload a photo. They called the app Instagram and launched it in October 2010 [10].

Creating new products or services is quite challenging because there is a high risk of creating something that nobody wants [4]. More than half of all product ideas do not work. Typical reasons are that customers are not excited about a product or that a product is too difficult or time-consuming to use. Sometimes, there are problems with the business viability due to legal, financial or business constraints [11]. Many assumptions are made during product development that come from team members or superiors. Product teams, for instance, try to take the customers perspective and they imagine that customer use a product in a specific way. When they observe real customer behavior afterwards, they are often surprised that customers behave quite differently. Due to Gladstone [9], "it is often the unexpected way that a customer uses a product, that highlights it is true potential".

In order to raise the odds of success of product and service development it is important to identify the important assumptions that need to be true for success. These assumptions need to be validated as early as possible. An important task is to identify these assumptions. But how to find them? Where are they documented? Usually, all relevant aspects of a product or business idea are documented in canvas models such as the Business Model Canvas [3] or the Lean Canvas [4]. At the beginning, these canvas models are full of untested assumptions. Therefore, canvas models can be seen as a good starting point for identifying assumptions.

Every entry in a business model is an assumption until we have proven that the assumption is right. Assumption testing is an essential activity [13]. However, product development is limited by time and other resources so that not every assumption can be tested. This is the reason why we should first identify which assumptions are the riskiest ones and test them first. Ries states that "Lean Startup is designed to operate in […] situations where we face […] extreme uncertainty…" [1]. Ries calls the riskiest assumptions "Leap-of-Faith Assumptions" (LOFA). They can be seen as claims in a business plan that will have the greatest impact on its success or failure. Very often, LOFAs focus in the beginning around the problem and the customer segment. Testing these assumptions is quite difficult as customers "often think they know what they want, but it turns out that they are wrong" [1]. Careful validation techniques, e.g. through customer development interviews, is necessary to validate those assumptions.

There are many methods dealing with risk prioritization, but there are only little research and practical experience on teaching them. In this paper, we describe a workshop format that guides participants on how to prioritize assumptions of an example business models or business ideas.

The rest of this paper is organized as follows: Section 2 presents related work. Section 3 defines the research approach and the research questions. In Section 4 we present the results followed by Section 5 with a discussion and lessons learned. Section 6 summarizes the paper and outlines future research.

2 Related Work

There exist several different approaches for prioritizing assumptions with respect to risks. In this section, we describe some of the popular methods. The methods have several differences: some are, for instance, using matrixes with dimensions, others are based on quantitative risk calculations, and some methods recommended specific sequences in which assumptions can be tested to reduce risks.

The first matrix approach is the Prioritization Matrix by J. Gothelf and J. Seiden [5]. They use the two dimensions “known to unknown” and “low risk to high risk” in order to classify and compare different assumptions. The second matrix approach is the Prioritizing Leap-of-Faith Assumptions (LOFA) matrix described in the book “The Startup Way” by Eric Ries [1]. Ries also uses two dimensions. The first one is the “time of impact” which describes when the assumption will have an impact. The second one is the “magnitude of impact” which describes how big the impact is on the business model if the assumptions are false [1].

The approach by J. Fjeld consists of a calculation with three parameters: severity, probability and cost. After all the parameters are calculated for every assumption, they can be ranked [6]. Ash Maurya divided risk into three different categories: product risk, customer risk and market risk. He recommends to prioritizing the assumptions based on the stage of your product [4]. The book “Value Proposition Design” by Alexander Osterwalder et al. [7] includes also a description of a simple prioritization method: a long line from the bottom “less critical to survival” to the top “critical to survival” is used for prioritization. The hypotheses from a business model can be pinned along this line and ranked in order [7]. Another method is described in the book “Disciplined Entrepreneurship” by Bill Aulet [12]: he recommends making a list of all the areas in which logical conclusions have been made, such as conclusions about producers, consumer and development. Laura Klein presents a method in her book “Building Better Products” [14] that is based on a risk identification grid with two separate factors. The first factor describes how likely it is that an assumption is true and the second factor is how bad the outcome will be if it is not true. For further details we refer to a previously published more comprehensive analysis of risk prioritization methods that has been conducted by the authors of this article [8].

3 Research Approach

In order to teach how to prioritize Lean Startup assumptions and to raise assumption thinking we created a workshop format that uses one of the aforementioned prioritization methods. For the workshop we selected as initial method Eric Ries’ method from “The Startup Way” which is based on sorting assumptions along the dimensions “time to impact” and “magnitude of impact”. The assumptions are mapped onto a matrix. All assumptions in the top right quarter, which have a high magnitude and a near impact can be seen as LOFAs. They should be tested with experiments as early in the product development process.

In the workshops, for the specific task of risk prioritization we wanted to teach the participants how they can easily classify assumptions into a risk matrix and identify the riskiest ones. We prepared original Airbnb assumptions, so the participants did not have to make their own assumptions initially. There are 22 assumptions in total that the participants worked with. All participants were divided into groups of 3 to 6 participants. At the beginning of the workshop we gave a presentation motivating the relevance of the topic and explaining what assumptions are in the context of Lean Startup. Additionally, we showed them some examples how startups identified and tested their assumptions. Directly before the risk prioritization task, the participants got a short introduction about Airbnb.

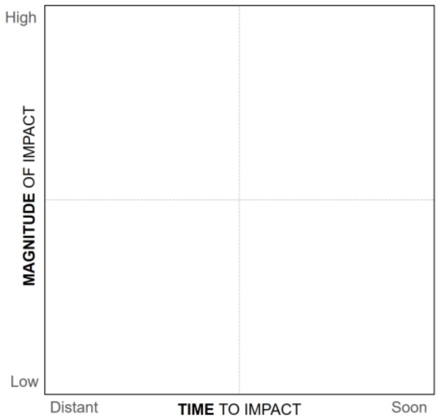


Fig. 1. - Leap-of-Faith Assumptions Matrix

(© Eric Ries - The Startup Way p.93)

After that, we showed them the task and explained the two axes of the Leap-of-Faith Assumption matrix. Additionally, we explained to the participants that the riskiest assumptions go in the upper right corner of the matrix where the distance from each axis is the greatest. During the task, the participants worked completely alone without any help. Each group got a poster with the assumption matrix. The assumptions were already written on prepared sticky notes. The participants could look at all the assumptions and potential relationships between different assumptions and decide where to put them in the matrix. Each group had 20-25 minutes for this task. After the task, the results were photographed with a camera and discussed. Each group was allowed to present the three to four riskiest assumptions they identified.

With this approach the following research questions should be answered:

- RQ1: Did the teams identify the riskiest assumptions?
- RQ2: Which assumptions are particularly correct/wrong categorized?

4 Execution and Analysis

A total of 6 workshops was carried out with 19 teams and in consequence a total result of 19 matrixes. Immediately after each workshop task, we captured the results in pictures. All pictures were copied to a digital folder and then individually printed on pages. After printing, we measured the length of each axis in the bottom left quarter. We used the results to scale the manually measured points to full-scale. Then, using a ruler, we manually measured the distance from each axis to the sticky notes with the assumptions. All the information from the measurements were recorded in Excel. Subsequently, we measured the same quarter on the bottom left of the original poster, took the value and set it in relation to the previous manual measurements. With this value, we scaled up all the manually measured points to the original size. The gained data was used for the analysis of the team results. In total, there were 397 sticky notes with 22 different assumptions.

4.1 Findings

Figure 1 shows the distribution of the assumptions in the prioritization matrix. The groups classified the assumptions into two different dimensions. The first dimension was “Time to Impact” and the second was “Magnitude of Impact”. Each dot represents one of the 22 assumptions. The position in the matrix represents the average positioning of the 19 groups for each assumption with respect to these two dimensions.

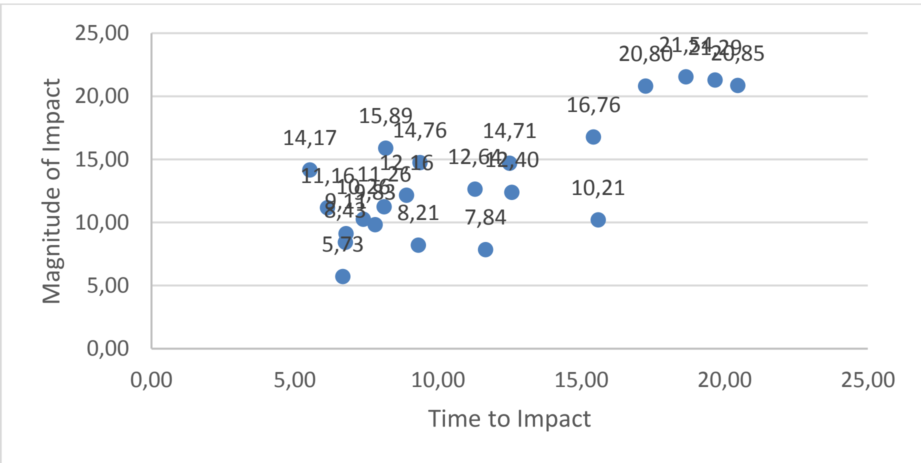


Fig. 2. - Distribution of all assumptions in the prioritization matrix

Table 1 presents the individual assumptions together with the average prioritization result for each individual assumption along the two dimensions. In addition, the standard deviation indicates the degree of agreement between the different teams with respect to the prioritization.

Table 1. – Prioritization results for each assumption and standard deviation

Id	Items	Assumption	TIME Ø	STD-ABW	MAGNI-TUDE Ø	STD-ABW
1	18	Travelers are willing to rent from strangers (no hotels)	**20,46**	4,53	**20,85**	2,73
2	18	Homeowners will allow strangers to live with them for a short time	**19,67**	3,34	**21,29**	2,29
3	18	AIRBNB is legal	**18,65**	4,82	**21,54**	2,63
4	19	There are enough travelers who accept the offer	**17,25**	5,92	**20,80**	2,31
5	19	Travelers seek alternative experiences to hotels	**15,42**	4,53	**16,76**	5,13
6	19	Adding pictures to descriptions results in a higher booking rate	**12,50**	6,8	**14,71**	5,3
7	16	Hotels are perceived as impersonal	**15,59**	3,84	**10,21**	5,89
8	16	Travelers find hotels too expensive	**12,58**	6,36	**12,40**	6,35
9	19	Displaying apartments on a map will result in more bookings	**9,37**	4,76	**14,76**	5,46
10	8	An indication of the ratings below the description of the apartments increases the likelihood of a booking	**8,17**	4,59	**15,89**	4,4
11	19	The cleanliness of the apartments is the most important thing for the travelers	**11,29**	5,92	**12,64**	4,37
12	19	It is very important for the travelers to check-in to the rented apartment at any time	**8,90**	4,88	**12,16**	5,6
13	19	Design A of the search page leads to more bookings like Design B	**5,53**	5,52	**14,17**	6,77
14	18	Travelers do not want to clean up after their stay	**11,66**	6,91	**7,84**	4
15	19	Through a Google Adwords campaign we get more bookings	**8,12**	5,58	**11,26**	5,24
16	19	The faster the search results appear, the higher the probability of a booking	**7,39**	5,11	**10,26**	4,53
17	19	Profile pictures of the travelers increase the likelihood that the landlords will accept the booking	**7,81**	5,12	**9,83**	5,39
18	19	Travelers like to pay for cleaning	**9,32**	5,15	**8,21**	4,06
19	19	Pictures by professional photographers lead to a higher booking rate	**6,14**	4,38	**11,16**	5,61
20	19	By sending vouchers to customers we get more recurring bookings	**6,79**	3,75	**9,11**	5,09
21	19	In the result of the apartment search it is better to see more than 5 hits	**6,76**	6,29	**8,43**	4,7
22	19	The price selection is easier with a sliding scale	**6,68**	4,49	**5,73**	3,2

Overall, the common agreement on the “Magnitude of Impact” dimension was greater than on the “Time to Impact” dimension.

In the “Time to Impact” dimension, the groups identified the following three assumptions as the most important: "Travelers are willing to rent from strangers (no hotels)", "Homeowners will allow strangers to live with them for a short time" and "AIRBNB is legal". The smallest standard deviation of the dimension was $\sigma = 3,34$ with the assumption "Homeowners want to allow strangers to live with them for a short time". The largest standard deviation was the assumption: "Travelers do not want to clean up after their stay" with a value of $\sigma = 6.91$.

With respect to the “Magnitude of Impact” dimension, the groups identified the following three assumptions as the most important: "AIRBNB is legal", "Homeowners want to allow strangers to live with them for a short time" and "Travelers are willing to rent from strangers (no hotels)". The smallest standard deviation of the dimension was $\sigma = 2,29$ with the assumption "Homeowners will allow strangers to live with them for a short time". The largest standard deviation was the assumption "Design A of the search page leads to more bookings like Design B" with a value of $\sigma = 6.77$.

One of the interesting results was that all groups from every workshop had always independently identified the same three riskiest assumptions.

4.2 Threats to Validity

In this section, we critically discuss our study results regarding internal and external threats to validity:

How comparable is the business model to other business models? In the selected example, we are dealing with a platform business model. Although, many business models fall into this category of business models, other archetypes of business models such as direct business models exist and this might impact the results.

The criteria “Time to Impact” and “Magnitude of Impact” were chosen as prioritization criteria. Other criteria can also play a role, such as the effort involved in testing. We chose this risk matrix as an initial approach to prioritize assumptions because it is proposed by Eric Ries popularized the Lean Startup approach.

Can the method we described be used outside of a workshop? We have tested the method and overseen its use in workshops. That does not necessarily mean the method works online.

Did the teams understand the prepared assumptions and were they clearly formulated? The teams explained to each other how they understood the prepared assumptions and ended up with a common vision. Two times, teams asked for the meaning of an assumption because they did not understand it correctly.

Is our evaluation correct? Were the results well photographed and are they usable? We tried to photograph the group results from a direct position as best as possible. The results were printed out on A4 paper and measured manually with a ruler. Some small inaccuracies remain. Firstly, the sticky notes had no exact reference point so we had to choose them freely. Secondly, some of the sticky notes were overlapping, making it difficult to set the reference points. The calculation was carried out with the help of Excel and was additionally controlled by another researcher.

What kind of prior knowledge did the participants have to bring along for the described part of the workshop? The participants needed to know the Airbnb business model to understand the assumptions. At the beginning of the workshop, we first asked whether they knew the business model or not. We then briefly explained what Airbnb does and we placed an Airbnb info sheet on each of the group tables. How were the appropriate Airbnb assumptions selected? The assumptions were made by the scientists using various sources of literature. Together, we selected the assumptions for the workshop and the selection was subjective.

Are the assumptions simply unfounded? Attention was paid to ensure that the assumptions were understandable, therefore other scientists were shown the assumptions and questioned if their meaning was clear.

Are the assumptions too simple and do not represent real assumptions? We extracted the assumptions from real Airbnb reports and books so we believe that the assumptions can be thoroughly tested.

In order to generalize the results, further research with more workshops and training is necessary.

5 Lessons Learned and Discussion

Overall, the workshops were well received and the participants had no major problems in conducting the tasks. The following lessons learned could be identified:

Providing an example case with a set of predefined assumptions seems to be an easy and efficient way to teach the concepts. This worked very well for the participants. The participants could get immediately involved in the task of risk prioritization and did not have to spend much time for coming up with their own assumptions. Additionally, the prepared assumptions had the advantage that the participants had no personal feelings about them so that they could view the assumptions more objectively.

The groups were randomly created, so that they typically consisted of participants with different backgrounds. The participants learned that there were different opinions on where to put the assumptions in the matrix and needed to come up with an agreement. The participants in each group were able to get a common understanding. Usually, the groups needed 20-25 minutes to map all assumptions on the matrix. In one workshop the group size was bigger, i.e., 5-6 persons per group. In this case the mapping took around 35 minutes.

If a team struggled with the classification of an assumption on the time dimension, it helped to give them a hint: "Think about the following: Which assumption needs to be successfully tested first". This helped the participants to better arrange their assumptions on the time dimension.

The two dimensions were quickly understood and there were rarely questions about the dimensions.

Working with the assumptions was fun for the participants and they gained a newfound awareness that identifying, understanding, prioritizing, and validating important assumptions is a highly relevant activity. After the task, some participants recognized that working with assumptions and testing the riskiest ones can also create significant effort.

After the workshops, the participants often asked if they could take the risk matrix poster and the used material home. This indicates that the participants have understood the importance of risk prioritization and that they are interested in applying this method to their very own business and product ideas.

6 Outlook

We plan to make improvements to the workshop materials so that a simpler and even more accurate analysis of the results is possible. The sticky notes will be provided with a reference point and a number in order to better measure the exact position in the matrix and to better support the analysis. We are also planning to conduct short qualitative interviews after the group work in order to complement the analysis. Further workshops are planned to increase the significance of the results.

Another research avenue we are currently discussing is to develop software-based simulators so that participants can learn prioritization online and/or by using more than one prepared business scenario.

References

1. Ries E.: The Startup Way. , Portfolio Penguin, 2017, pp. 89-93, p. 99
2. Cooper B., Vlaskovits, P.: The Lean Entrepreneur, Second Edition, Wiley, 2016, pp. 127
3. Osterwalder A., Pigneuer Y., Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers, Wiley, 2010

4. Maurya A.: Running Lean, 2012, Second Edition, O'Reilly
5. Gothelf J., Seiden J.: Lean Ux Applying Lean Principles to Improve User Experience, 2013, First Edition, O'Reilly, pp. 22
6. J. Fjeld: How to Test Your Assumptions, MIT Sloan Management Review, Winter 2018, pp. 89-90
7. Osterwalder A., Pigneur Y., Bernada G.,Smith A. :Value Proposition Design, Wiley, 2014, pp. 203
8. M. Gutbrod, J. Münch and M. Tichy : "The Business Experiments Navigator (BEN)", IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC) 2018 vol 18
9. B. Gladstone: "https://www.codegent.com/blog/2014/3/assumption-testing", Codegent Blog 2014
10. K. Sawyer: Zig Zag: The Surprising Path to greater Creativity, Wiley, 2013, pp 22
11. M. Cagan: Inspired - How to create Tech Products Customer Love, Wiley, Second Edition, 2018, pp 111
12. B. Aulet: Disciplined Entrepreneurship, Wiley, First Edition, 2013, pp. 219
13. J. Liedtka and T.Ogilvie: Designing for Growth, Wiley, First Edition, 2013, pp.131
14. L. Klein: Building Better Products, Wiley, First Edition, 2016, pp. 245-247