

Minimum Viable Product or Multiple Facet Product? The Role of MVP in Software Startups

Anh Nguyen Duc^(✉) and Pekka Abrahamsson

Department of Computer and Information Science (IDI),
NTNU, 7491 Trondheim, Norway
{anhn, pekkaa}@ntnu.no

Abstract. Minimum viable product (MVP) is the main focus of both business and product development activities in software startups. We empirically explored five early stage software startups to understand how MVP are used in early stages. Data was collected from interviews, observation and documents. We looked at the MVP usage from two angles, software prototyping and boundary spanning theory. We found that roles of MVPs in startups were not fully aware by entrepreneurs. Besides supporting validated learning, MVPs are used to facilitate product design, to bridge communication gaps and to facilitate cost-effective product development activities. Entrepreneurs should consider a systematic approach to fully explore the value of MVP, as a multiple facet product (MFP). The work also implies several research directions about prototyping practices and patterns in software startups.

Keywords: Prototype · MVP · MFP · Software startups · Software development · Empirical study · Exploratory case study

1 Introduction

Software industry has witnessed a growing trend of software products developed by software startups, often newly created companies with little operating history aiming at high-growth software products. Different from established companies, startups typically deal with identifying and implementing a product that delivers actual customer value [1]. Recent methodological approaches for startup product development, i.e. Lean startup [3] or new product development processes [2] emphasize the ability to learn about actual problems from early customers and the speed of learning. According to Lean Startup [3], every startup should start with building a Minimum viable product (MVP), and use it to validate their hypotheses about customer needs.

MVPs, defined as products with just enough features to gather validated learning about the products, is a major focus in early stages. It plays an important role not only for a startup team, but also the startup's external stakeholders, such as potential users, investors and mentors. Nowadays, MVP is a key artifact to be shown in a meeting with an investor. There are several different types of MVPs, varied by development efforts, their purposes and stages they often occur [3]. For instance, a landing page, as one

MVP, can be quickly created to communicate the product proposals to public. A single feature prototype, as another MVP, might take several months for construction and integrate into final product. Besides, different MVPs might be used to serve the same purpose, for instance, to communicate with investors.

It is little known about how MVPs are used after their creation, from both community of practitioners and researchers. Given the importance of MVPs for early stage startups, we are interested in understanding how the MVP is used in software startups:

“RQ: How are MVPs used in early stage software startups?”

We argued that from an engineering perspective a MVP shares a lot of characteristics with a software prototype. Prototyping has a long history in Software Engineering (SE) research, as an essential part of water fall life cycle [5]. However, in SE research, there is little discussion about prototypes in the context of software startups [6, 7]. In this paper, we discussed about the usage of MVP in the relation to prototype’s characteristics. We also argued that MVPs has been used to communicate with external stakeholders, such as investors and early customers. Information System (IS) has a theory to explain about how an artifact was used to communicate among different communities with different expertise [8]. Therefore, we utilized the boundary spanning theory to initiate and to capture the MVP usage.

The paper is organized as follows; firstly we presented backgrounds about MVPs, software prototype and boundary spanning theory (Sect. 2). Then, we described our research approach and case description (Sect. 3). After that, the qualitative findings are presented (Sect. 4). Finally, we discussed the reflections of study, threats to validity (Sect. 5), conclusion and future work (Sect. 6).

2 Background

2.1 Classification of MVPs and Prototypes

Eric Ries initiates the classification of MVP types [3], which are discussed among the community of practitioners, including:

- Explainer video: a short animation that explains what your product does and why users should buy it. The video is often simple, lasts for 30 s to few minutes.
- Landing page: a web page where visitors “*land*” after clicking a link from an e-mail or another type of a campaign. A landing page is used to quickly communicate the startup proposals, to diffuse objections, and to call the visitors to action.
- Wizard of Oz: an user interface that looks like a real working product, but the actual business process is manually carried on. The purpose of this MVP is to demonstrate the complete job done by the product.
- Concierge MVP: a manual service that consists of exactly the same steps users would go through with the product.
- Piecemeal MVP: similar to Wizards of Oz MVP, however, execution of the tasks is done by using existing tools.
- Mockup MVP: such as, paper prototype and wireframe, was representative of product user interface without any functionality.

- Public project proposal: Kickstarter and other crowdsourcing sites allow for users to pre-purchase the product and provide a great way to raise money for initial orders.
- Single feature MVP: a prototype that implements the most important function of the product.
- Rip off MVP: a successful product to get feedback, then pivot in a different direction.

The term “prototype” is also often used in startup context as an interchangeable term with MVP. There are different types of software prototypes often used in early phases of software development, such as throwaway, or rapid prototype, which consumes very little efforts with minimum requirement analysis to build a prototype [9]. Another type of prototype is evolutionary prototype, which bases on building actual minimal functionality in the beginning [9]. Last but not least, incremental prototype refers to building multiple functional prototypes of the various sub systems and then integrating all the available prototypes to form a complete system [9]. In this paper, we use the above categories to differentiate and discuss about different type of MVPs during early-stage software startups.

2.2 Theory of Boundary Spanning

To explain the roles of MVPs and prototypes, we borrow the theory of boundary spanning across boundaries in software startups. From the view of knowledge management, most innovation happens at the boundaries between specialized pools of knowledge [8, 10, 11]. Three types of knowledge boundary is commonly mentioned in IS literature:

- A syntactic knowledge boundary occurs when there is a lack of a shared syntax and creates the concern that information may not be processed properly across a given boundary [8]. For instance, entrepreneurs use business terms that make developers do not understand.
- A semantic knowledge boundary occurs when a common syntax is present, different interpretations of the common syntax make communication and collaboration difficult [8]. For instance, a designer might think about artistic mindset while a developer think of software architecture when talking about design thinking.
- Pragmatic knowledge boundary occurs when a common interest has to be achieved when participants negotiate with each other on the scope [8], consequences and conflict solutions of knowledge delivery, i.e. developers and entrepreneur do not share common interests, i.e. a clash of interests occurs.

Boundary artifact is used to cross these different types of knowledge boundaries [10]. The theory states that an artifact only helps bridging knowledge boundaries if it qualifies as a boundary object, which is described as an artifact that “*sits in the middle*” of diverse knowledge groups, establishing a “*shared and sharable*” context for distributed problem solving. These artifacts need to be “*both plastic enough to adapt to local needs and constraints of the several parties employing them, yet robust enough to maintain a common identity across sites*” [11].

3 Research Approach

3.1 Study Design and Case Selection

We conducted this study by using a multiple-case study design [12]. As shown in Fig. 1, we adopted a mixed approach of deductive and inductive research. The initial observations about MVP usage were extracted from Case B and abstracted by using classification from software prototyping and theory of boundary spanning. The initial themes were used to guide the analysis of interview transcripts later. The final thematic scheme of significant MVP usage was extended from all five case studies.

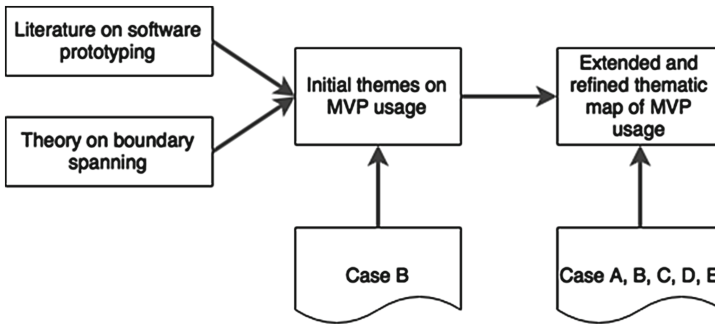


Fig. 1. Research approach

These cases describe startups from the seed-stage to the early growth-stage i.e. from ideas to prototypes and operating products. For concealment the startups are not named in this paper, but are instead referred to as Company A, B, C, D and E, as described in Table 1. The cases are selected by using our industrial network, using three selection criteria: (1) companies have at least three people and first paying customer, (2) companies have at least six months operations, (3) and companies have performed some types of software development. The industry domain varies from retail, marketing to construction. Cases come from Italy, Norway and Finland with company size vary from three to 18 full-time employees. Most cases have been operated mainly by self-funding. Business models include both Business-to-Business (B2B) and Business-to-Customer (B2C). All of the investigated startups were founded by experts in software development.

3.2 Data Collection and Analysis

Methodological triangulation in data collection is implemented by using documents, interviews and observation. Business documents, such as business model canvases and full description of business plan was exposed to the research team as a preliminary step prepared for interviews. Interview is our primary source of information. In most of the cases, we conducted multiple interviews with their CEOs, CTOs and co-founders. The interviewees were asked questions about (1) realization of business idea (2) pivot practices (3) product design and development. Observation is useful to understand how

MVPs and prototypes were implemented and used in the working environment. In Case B, the author participated in five weekly meetings. In Case A, the CEO has provided a narrative description of the startup process and observations from that.

Table 1. Startup case demographic

Id	Product	Year	Loc.	Dev. approach	# Ppl.	Latest Stage
A	Online photo marketplace	2012	Italy	Lean startup, Tailor Agile	6	Implementation
B	Marketplace for food hub	2015	Norway	Adhoc	3	Conceptualization
C	Collaboration platform for construction	2011	Norway	Distributed Scrum	4	Commercialization
D	Sale visualization	2011	Norway	Tailor Agile	18	Commercialization
E	Under water camera product	2011	Finland	Adhoc	3	Implementation

We used thematic analysis to analyze the data, a technique for identifying, analyzing, and reporting standards (or themes) found in qualitative data [13]. We started by reading all interview transcripts and relevant documents, and coded them according to open coding [14]. Each segment of text that expresses MVPs and the usage of these MVPs or prototypes were labeled with an appropriate code. The MVPs were later classified into the MVP types, prototype types and boundary spanning types, if relevance. The emerged MVP usages were compared across interviews and finally merged into a final thematic map.

4 Result

4.1 Types of MVPs

Table 2 summarizes different types of MVP used in our cases. According to the data, software startups adopted several types of MVPs in early stages. Landing page were used by all cases, often during the product development or close to the product launch. Different types of mockups were used extensively during early stages. For example, Case B used a wireframe tool called JustInMind, as the major tasks in the beginning of their project. In Case C, paper prototypes were used during most of all customer meeting. Except Case C, all of our cases started early with developing the first most important feature of their product. Other types of MVPs, such as Concierge MVP, Wizard of Oz and Picemeal MVP were also used in some cases. In the next sections, we described three main roles of these MVPs, which are design artifact, boundary spanning object and reusable artifact.

Table 2. Prototyping approaches in our cases

	Cases				
	A	B	C	D	E
Types of prototype					
Landing page	X	X	X	X	X
Mockup MVP	X	X	X	X	X
Single feature MVP	X	X		X	X
Concierge MVP				X	
Explainer video			X		
Wizard of Oz		X			
Piecemeal MVP			X		

4.2 MVP as a Design Artifact

Table 3 describes the themes that were grounded from interviews, As a design artifact, a MVP facilitates the visualization of ideas, the reflection on architectural design and the innovation process.

Table 3. Data grounded themes on prototype usage

	Companies				
	A	B	C	D	E
MVP as a design artifact					
Visualizing design idea	X	X	X	X	X
Reflection on architectural design		X	X		X
Facilitation of creativity		X		X	
Clarifying mismatches on user expectation	X	X			X
MVP as a boundary spanning artifact					
Bridge between Business mind vs. Technical mind	X	X			X
Bridge between Entrepreneur team vs. End user		X	X		
Bridge between Entrepreneur team vs. Investors			X	X	X
MVP as a reusable artifact					
Documentation	X	X	X		
Growth hacking mechanism		X		X	
Bootstrapping tool	X	X	X	X	X

Visualizing Design Idea: As a rapid prototype, MVP is a mean to travel from idea to real product. In Case B, paper-based UI prototypes were used during brainstorming sections when the team virtually meets. The CEO mentioned, “*Each of us has our own design version of [Product name], when [CTO name] describes his idea about sharing meals among students ... We start sketching the workflow and the app UI right away ...*”.

The practice is also found in Case C and D, for example, “*During the design meeting, the team worked together in the a collaborative mockup prototyping tool. The team members continued giving inputs to refine the prototype.*”, mentioned by the CEO of Company C.

Initial ideas and prototypes can vary, hence, cross-check during prototyping phase is often necessary. For non-technical founders, visualizing their thoughts is important to provide inputs for technical design: “*I have many great ideas, but I have no idea if they can be implemented. Building a prototype at least allows me and also others in my team to ask the right questions... Visions and theory are notoriously hard to implement. A prototype has to be real enough to be convincing, without looking like science fiction.*” (CEO of Company C).

Reflection on Architectural Design: MVP prototyping process is where product design is reflected and revised. In Case B, mockup MVPs were created by the CEO to capture the idealization phase. Meanwhile, the architecture of a product was initiated by the CTO. The mockup MVP and architectural design was started at the same time and gradually became two separate tasks that reflect business requirements and technical insights. After talking to early customers, the MVP was updated according to new requirements. Consequently, the MVP became a batch of new inputs for the final product architecture: “*From looking at the MVP you can see that the options for taken-away or eat-with-host is not there in our workflow. I will update it in the next meeting ...*” (CTO of Company B). It is also similarly mentioned in Case E, while the CTO reflected on how they had changed the code structure based on early feedback from early stage working prototypes: “*Refactoring is not too big an issue compared to benefits of early releases...*” (CTO of Company E).

Facilitation of Creativity: MVP, as a rapid prototype, is more important than idealization phase, as it gives the balance between realistic and futuristic design. In Case B, the process of finalizing a product idea has a typical path of a new product development process [2]. Several ideas were discussed from the beginning, such as mood tracking, event scheduling, e-receipt and food sharing. After many internal discussions, the focus is to create a platform that facilitates gathering with friends by sharing food. Diverged from theory, idea screening and concept testing was not really distinguished and occurs iteratively in Case B. As ideas could come from all team members, to illustrate a given concept, the CTO created a small prototype to convince other team member. From experience of a serial entrepreneur, making a concrete visualization of an idea will make his/herself and other team member easier to evaluate the innovative characteristics of the product: “*When initiating in my mind, the idea sounds great. When putting it into paper, it looks similar to existing products that I know.*” (CEO of Company D).

Realize Prototype-User Expectation Mismatch: MVP is also appeared as a part of Lean startup approach to adjust the problem-solution fit. Some MVP, i.e. single feature MVP is the latest point in time where disagreement, misalignment and different perspectives are harmonized for the sake of the project success. For example, in Company E, the CEO mentioned: “*Real-life use cases give always nasty surprises compared to the lab environment. In my case, river-side installations in our case are fairly challenging. The deployed version gives much lesson to learn*”.

4.3 MVP as a Boundary-Spanning Object

The interview data revealed that MVPs facilitate bridging knowledge gap between the entrepreneur team and external stakeholders, i.e. customers, mentor, vendor and investor.

Bridge between Business mind vs. Technical mind: MVP is used to communicate about technical detail and business idea, which often is the case of early stage startups. In Company B, a syntactic boundary occurred during an early stage of team formation by a lack of the consistent use of technical terms. A mockup MVP was used to facilitate common language: *“She is very sharp about business and finance stuffs, but it takes a long discussion to explain her about the importance of having flexible product design ...”* (CTO of Company B). The gap also occurs in case the product is technically complicated, as described in Company E. Technical details was too much to verbally explain in our interview, which can lead to a threat of synaptic knowledge gaps. The CTO decided to use a paper architectural diagram to hide some of the technical details, but still convey the product ideas and good level of technology. In Company A, we found a quote presenting a semantic knowledge boundary between the CEO and a developer: *“I asked the guy (developer) to create a registration page and he has done a complicated page with all the detail ... I only need a very simple login function ...”* The CEO mentioned that if a paper description was given, the mis-interpretation might not be there.

Bridge between Entrepreneur team vs. End users: As mentioned in [3], MVP is used to validate if the entrepreneur’s ideas are the same with end user’s expectation. In Company B, the idea was to develop a platform for sharing food and food-based social gathering. Presenting the ideas to people without showing a MVP was quite difficult: *“We have done interviews with some friends ... by explaining key concepts like cuisine, Airbnb of food, ... which is not effective”* (CEO of Company B). Rapid prototypes, such as landing page and explainer video were proposed to communicate to a large amount of audiences: *“As a suggestion for the next entrepreneurs, one things we should do from the beginning is to create a landing page. It is always difficult to follow up after interviews if you do not have a link for them”* (CTO of Company B).

In Company C, the product serves for construction tenants, the CEO had stayed in customer organization for a period to understand gaps in the current work culture and process. At the beginning, without a MVP, the CEO had a hard time to convince customers about the benefits of her product. Syntactic knowledge gap was the barrier when the CEO needed to learn about their language. The one-feature MVP was used to show practical use of her solution: *“We work with a customer organization, learn how they have worked with the current solutions and describe our proposal via the prototype. It is hard for them to realize the benefit without concrete examples...”* (CEO of Company C).

Bridge between Entrepreneur team vs. Investors: knowledge gaps were observed not only within internal members, but also between entrepreneur teams and external stakeholders, such as vendor and investors. MVPs were used in Case E to support communication and negotiation beyond the team boundary: *“A three-dimensional*

prototype is always better than just a documented specification when negotiating contracts for manufacturing, support, and marketing. As a startup, you need all the leverage you can get” (CTO company E). In Company C, MVPs were used to reduce misunderstanding between entrepreneur team and outsourcing vendor. The CEO of Company C mentioned that mockup MVP is the major mean to communicate with the development team in India: *“I can’t seat here and write about hundred page features and that not sure everyone understands.”*

Observations from investor pitches in Company B suggest that MVP is always recommended in any pitches and be a part of evaluation criteria. This is also mentioned by CEO of Company D: *“It is important to show investors that you are committed, and past the idea stage. Without a prototype, most professional investors won’t take you seriously.”* While most of investors have certain knowledge about technology and the domain, the threat can be eliminated is the pragmatic knowledge gap. The presentation can be more interesting with demonstration, and attracting interest of investors.

4.4 MVP as a Reusable Artifact

Aligned with bootstrapping approaches of many software startups, MVPs need to be useful in many purposes. Even for a throw-away prototype, it can be used later in the startup processes for other purposes.

Documentation: MVP is a way to document project progress and technical documents. In Company B, a wireframe is implemented using JustInMind, with concepts of layers, reusable objects and screen scenarios. The tool also provides a function to generating html versions with textual descriptions. In Company C, single feature MVP is made in a self-explained and changeable manner. It is also included architectural decisions and instruction for further extension. Besides, each prototype is an important milestone marker to quickly keep track on pivoting: *“it doesn’t matter how certain you are about your solution; it probably will take several changes soon. It’s much easier to pivot the pre-production prototype than to dispose of unsellable inventory... We can later understand why we have changed from that prototype.”*

Growth Hacking Exploration: Prototyping is the phase where growth hacking can be experienced. Growth hacking techniques help to increase the amount of users, often require the knowledge about both marketing and software development. In Company B, one of the early discussions was on what type of MVPs should be used in the current stage. After consulting with mentors, the team decided to use a mockup MVP that is hosted in a public server for having better reach: *“We decided to use a mockup MVP, it is hosted in Google web server. The link was attached to our online questionnaire so we can reach more people than going to each individual interview”.* In Company D, video was used in early stage to explain the concepts to large amount of customers without going into detail of sale and marketing terms. When the first one-feature MVP is available, it is freely offered to some organizations as beta testing.

Bootstrapping Mechanism: MVP is an economical approach of having a product, which is demonstrable to investors and early customers. In Company E, both software

and hardware technology is needed in the product. They adopted multiple iterations to gradually improve quality and performance of the product. It is mentioned that prototype reduces cost of final product development: *“One purpose of the long prototyping process is that we can better learn about the technology. Once technical uncertainties are clear, we can start again much faster with a clean product.”* (CTO of Company C).

For startup generally, time means wasting opportunities, and would be come competitor’s advantage. In Company D, the CEO suggested that *“Don’t spend your whole development budget, before finding that you need another iteration.”* Company D composed of all technical members from the beginning: *“You could say that we have followed the Lean startup, the first MVP we have at December 2012 when we was in [Incubator place] ... We focus on the development of the MVP from Day one”* (CEO of Company C). With heavily focus on product development, they implemented the strategies that making different prototypes of the same domain area. These MVPs later can be (partly) reused by integrating into another product. CEO of Company A mentioned: *“In reality, the process of designing, building, and validating a prototype does dramatically reduce the risk, and allows everyone to hone in on the real costs of going into production”*.

5 Discussion

As a central part of build-measure-learn, Eric Ries emphasized the main role of MVP as an artifact for customer validation [3]. Based on five case studies, we found that MVPs could be useful for a startup as a design artifact, a boundary spanning artifact and a reusable artifact. The process from business ideas to a launching product consists not only loops but also parallel branches. When market validation and product design tasks are carried on at the same time, certain types of MVPs would play a role of mutual adjustment between input from customers and product design. In many cases, we observed the benefits of having MVPs on final product development, such as increase feedback quality and reachability.

Adoption of MVP might be influenced by many contextual factors. We discussed about one most relevant factor, which is product development methodology. In our cases, Agile development is the most viable processes for software startups (in Case A, B, C, and D). In this context, fast releases with an iterative and incremental approach shorten the time from idea conception to production. The continuous integration might be the impetus for popular adopting evolutionary prototypes and single-feature MVP in our cases (Company A, B, D, E). However, the prototyping process might be hindered by other business and technical factors, leading to the inappropriateness of Agile principles sometimes during the startup process. In Company A and D, the evolutionary prototypes were implemented quite early and quickly during the process. While in Company E, the prototype is evolved gradually over months, due to the technical complexity of the product.

Reflecting on boundary spanning theory [10, 11, 13], we observed all three types of knowledge boundary within a startup team and also between the team and external stakeholders. MVP has been shown as an effective tool to break all these gaps. Syntactic knowledge boundary was found between the CEO and a customer when

explaining the product. It seems that syntactic boundary is not the main issues in our startup cases. The reason may due to the nature of products (for wide range of users in Company A and B), the familiarity of the CEO with the industry (Company C) and the familiarity with some customers in the field (Company C, D and E). Semantic boundary was found in a conversation between CEO and a developer, which can be observed more within an entrepreneurial team. We have not found much evidence about the boundary with mentors or investors, but it might happen as well. Pragmatic boundary seems to be the most important issues among the team members and between the team and investors. This can because of the divergence of entrepreneurial team in term of startup goals and motivation over time (Company A), or pitch and presentation skill to attract interest from investors and customers (Company B). We observe at least in these cases that MVPs play the role of bridging these gaps.

Our research revealed different ways that MVPs can be used to support startup business activities. However, they are not equally perceived among all startups. For instance, only in Case B most of the MVP usages were identified. Moreover, tactics with using MVP are arbitrary and there is no systematic approach to fully utilize the benefit of MVP. For practitioners, we suggested that the development of MVP should consider the ability to communicate among different stakeholders, to facilitate the design and to save business and product development costs. In short, MVP should be developed as a Multiple Facet Product (MFP).

There are several threats to validities worth to discuss [15]. One internal threat of validity is the bias in data collection, as data might not represent the comprehensive story. An important issue is related to the fact that the limited number of interviews might not represent the complete scenarios in our context of study. In order to mitigate this threat we selected CTO and CEO as interviewees, who have the best understanding about their startups. We also use other types of data sources to increase our understanding about the cases. With Company A and Company B, we also acted as the startup team members, which enables a lot of insights beyond interviews. Another internal threat of validity is about how reliable the reported cases are. This is ensured as two of the authors have not only theoretical background about software startups but also hand-on experience.

A construct threat of validity is a possible inadequate description of constructs. During the coding of interview transcripts, we adopted explanatory descriptive labels for theoretical categories, to capture the underlying phenomenon without losing relevant details. An external threat of validity is the representativeness of our selected cases. As discussed earlier, the sample is collected from European technical-founded bootstrap startups. A startup from USA or America, and a startup with different financial models might introduce other MVP usage patterns. For more thorough understanding and generalization of the results, a large pool of startups with variety profiles should be included. All of the cases are small startups under development seeking for seed funding. Besides, the startup decisions on MVP might be influenced by individual personalities.

6 Conclusions

The study of five startups reveals some insights for prototyping approaches in software startups. We found that MVP is also be used as a MFP, where it supports the design process, bridges communication gaps and facilitate the cost-saving activities. When market validation and product design tasks are carried on at the same time, certain types of MVPs would play a role of mutual adjustment between input from customers and product design. MVPs were used to bridge knowledge gaps between entrepreneurs and developers, customers, investors. Particularly, we illustrate how three types of knowledge boundaries have been resolved using MVPs.

So far, we have explored the stated research questions through a multiple case study. Our next step in the research is to include different types of software startups. Possible sequences of MVPs to be developed were initially observed in Company B and Company C. Future study will explore in-depth about MVP development processes in other cases. Another research topic is to understand how software prototype practices fit into Agile development context. Last but not least, OSS adoption has been essential for product development in startup. The question would be in which way startup companies can really benefit from adopting OSS.

Open Access. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, duplication, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, a link is provided to the Creative Commons license and any changes made are indicated.

The images or other third party material in this chapter are included in the work's Creative Commons license, unless indicated otherwise in the credit line; if such material is not included in the work's Creative Commons license and the respective action is not permitted by statutory regulation, users will need to obtain permission from the license holder to duplicate, adapt or reproduce the material.

References

1. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: A systematic mapping study. *Inf. Softw. Technol.* **56**(10), 1200–1218 (2014)
2. Coleman, G., O'Connor, R.: An investigation into software development process formation in software start-ups. *J. Enterp. Inf. Manage.* **21**(6), 633–648 (2008)
3. Ries, E.: *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses.* Crown Business, New York (2011)
4. Khurum, M., Fricker, S., Gorschek, T.: The contextual nature of innovation - An empirical investigation of three software intensive products. *Inf. Softw. Technol.* **57**, 595–613 (2015)
5. Sommerville, I.: *Software Engineering*, 9th edn. Pearson Education, Harlow, England (2010)

6. Bosch, J., Holmström Olsson, H., Björk, J., Ljungblad, J.: The early stage software startup development model: A framework for operationalizing lean principles in software startups. In: Fitzgerald, B., Conboy, K., Power, K., Valerdi, R., Morgan, L., Stol, K.-J. (eds.) LESS 2013. LNBIP, vol. 167, pp. 1–15. Springer, Heidelberg (2013)
7. Lindgren, E., Münch, J.: Software development as an experiment system: A qualitative survey on the state of the practice. In: Lassenius, C., Dingsøyr, T., Paasivaara, M. (eds.) XP 2015. LNBIP, vol. 212, pp. 117–128. Springer, Heidelberg (2015)
8. Carlile, P.R.: A pragmatic view of knowledge and boundaries: Boundary objects in new product development. *Organ. Sci.* **13**, 442–455 (2002)
9. Floyd, C.: A systematic look at prototyping. In: Budde, R., Kuhlenkamp, K., Mathiassen, L., Zullighoven, H. (eds.) *Approaches to Prototyping*, pp. 1–18. Springer, Heidelberg (1984)
10. Tushman, M.L., Scanlan, T.J.: Boundary spanning individuals: their role in information transfer and their antecedents. *Acad. Manage. J.* **24**, 289–305 (1981)
11. Bowker, G., Star, S.L.: *Sorting Things Out: Classification and Its Consequences*. MIT Press, Cambridge, MA (1999)
12. Yin, R.K.: *Case Study Research: Design and Methods (Applied Social Research Methods)*, 5th edn. SAGE Publications Inc, Thousand Oaks (2014)
13. Strauss, A., Corbin, J.: *Basics of Qualitative Research Techniques and Procedures for Developing Grounded Theory*. 2nd edition (1998)
14. Boyatzis, R.E.: *Transforming qualitative information: thematic analysis and code development*. Sage Publications, Thousand Oaks (1998)
15. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. *Empirical Softw. Eng.* **14**(2), 131–164 (2009)