

Yes, we can!

Building a capable initial team for a software startup

Pertti Seppänen

M3S/M Group
University of Oulu, FI 90015 Oulu, Finland
pertti.seppanen@oulu.fi

Abstract. Startup companies are based on the founders' innovations and visions of new business opportunities. Software startups are commonly considered as especially innovative. Besides the importance of the innovation and business vision, in the early stages of the startup the initial team plays a key role in transforming the innovation into a product or a service. At the same time, software startups are often small, immature companies with very limited resources. That highlights the importance of the initial team's capabilities to address the challenges of the product development from the innovation – the knowledge, experiences, skills, and other cognitive abilities.

In this chapter, we present the results of studies on the initial team's capabilities from the viewpoint of the product development, planning, designing, implementing and verifying the targeted product or service. The studies were conducted on a group of thirteen software startups in Italy, Norway and Finland.

The studies revealed that from a group of very heterogeneous software startups a generic structure of the initial team can be identified, consisting of three different roles, each having a specific set of responsibilities and capability needs. This team structure provides a software startup with a balance between the team's capabilities and problems and challenges to be solved during the early product development process.

In addition, we present the sources of the needed capabilities, the initial knowledge, experience, and skills of the founder, and broadening and deepening the initial capabilities by validated learning and by growth towards to the identified team structure.

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

1 Introduction

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

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

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

2 Prior research

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

2.1 Prior research on software startup challenges

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

Table 1. Internal challenges for the initial team.

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

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

Table 2. External challenges for the initial team.

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

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

2.2 Prior research on capabilities

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

2.3 Prior research on innovation ownership

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

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

3 Research design

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

3.1 Case selection and research data gathering

Studying a group of early stage software startups provided us with an insight what development-related capabilities are needed in software startups and give us a basis for the exploration how such an initial team is built and structured that has the capabilities necessary to tackle the challenges. Our study was based on a sample group of thirteen companies in four different geographical areas in Europe having a broad variation in product ideas, customer segments and utilized technology, as shown in table 3.

Table 3. Studied software startups

| Location | Product type | Customer segment |
|-----------------|------------------------|---------------------------|
| Italy | WEB application | Photographers, public |
| Norway | WEB application | event organizers, public |
| Norway | WEB/Mobile application | emergency centers, public |

| | | |
|------------------------|-----------------------------------|--|
| Finland, Oulu area | Instrument, embedded software | specific sector of public |
| Finland, Oulu area | Instrument, embedded software | specific sector of public |
| Finland, Oulu area | IOT device, embedded software | smart device vendors |
| Finland, Oulu area | Imaging system, embedded software | researchers |
| Finland, Oulu area | WEB application | nurseries, parents with small children |
| Finland, Oulu area | Instrument, embedded software | public |
| Finland, Oulu area | Software development services | systems and software companies |
| Finland, Oulu area | Special IT services | systems and software companies |
| Finland, Helsinki area | Aircraft maintenance software | aircraft carriers |
| Finland, Helsinki area | Graphical UI platform | smart device vendors |

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

3.2 Research data analysis

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

4 Results

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

4.1 Capability domains for product development

Though the details of the development-related capability areas varied between individual software startups, we could identify in our first thematic synthesis a high-level capability mapping with four main areas, as shown in table 4 [14].

Table 4. High-level technical capability domains of software startups

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

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

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

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

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

4.2 Team roles

In the second thematic synthesis round, we were able to identify three different personnel roles, each contributing to the overall capability level of the startups in a different way. The personnel roles are shown in table 5 [14]:

Table 5. Team roles

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

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

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

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

The experiences and skills of the hired developers, in turn, varied a lot in our study group. Being young organizations and having limited resources, the startups were forced to balance between the requirements set by the targeted products and the costs of developers with longer experiences in the field. The balancing between the needs and the economical possibilities was taken care of in different ways. Some companies hired only students who were assumed to be cheaper and more willing to work in startup-type companies, some invested in fewer but high-qualified professionals. In the former case, the founder herself had a solid experience and knowledge base on software

development and was therefore confident in being able to lead the work of less-experienced developers. In the latter case, the reasons to hire experienced developers were for instance the focus on a specific technology or especially high-quality requirements. Typical solutions were to hire old work-mates of the founder, or persons with otherwise known professional careers.

Even though the above explained way of hiring professionals to the positions of developers due to specific demands, we could identify besides the founder and the development team an additional personnel role in the software startups, the experts. Experts were persons who compensated for the capability shortages of rest of the team, especially the shortages of the founder. Mostly the experts' contributions were focused on the areas of the special technology domain or the process and quality domain (table 4), but in cases of founders without personal software development experiences the experts also compensated the founders' missing software capabilities.

Experts worked both as hired personnel and as sub-contractors. Sub-contracted experts were used especially in solving the problems of the specific technology areas needed in the product development [23]. Such needs appeared due to the fact that even experienced founders were sometimes less familiar with all the required technologies, because experienced founders tended to have new or more ambitious product views. In those cases the experts' contributions were very focused, both in terms of problems and time.

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

In the early stages, startups might experience different contractors for developing prototypes [23]. Outsourced tasks or sub-modules at these phases are small-scale and experimental. Given that, startup founders who lack of technical competence often choose outsourcing as a shortcut to a later stage of startups, where they can attract funding for proper product development. In some other cases, startup look for a sustainable strategy for product development, using their unique advantages, such as a personal relationship with a reliable outsourcing team, or successful collaboration previously [23].

4.3 Means to build the team capabilities

The third round of thematic synthesis was conducted to identify the means to acquire the capabilities in the initial team of a software startup. Though the details of how the teams were built up in the case startups varied, three high-level means that were common for all cases were identified, as presented in table 6 [13].

Table 6. Means to acquire team capabilities

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

5 Discussion

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

5.1 The founder – the initial capabilities

Our findings are along to the results of the prior research – the founder builds both the innovation and the initial team's capabilities on her personal capabilities, knowledge and experiences. Especially valuable a founder's prior experiences are in the application knowledge domain [19], laying the basis for both a successful business case and a smooth development of the product or service from the innovation [2, 19]. Examples of experienced founders are experts who have worked earlier in another, possibly bigger, company on the same business and technology areas, serial entrepreneurs, or persons gaining their knowledge through personal interests, such as contributions to open-source projects [13]. In those cases, the founder's own capabilities build a strong basis for the startup, the founder master the key areas of her new enterprise and the rest of the initial team is built typically to broaden the development-related capacity of the startup.

Cases of the opposite, founders without prior experience, are young persons who vote for entrepreneurship right after the graduation, persons that change their interest and future plans to some totally new area not known in advance, or persons that master the needed capability areas only partially, such as managers without own software development skills [13]. In those cases the founder needs a team that is capable in compensating for her shortages, whether they are in the application domain, technology domain, business domain, or any necessary work domain in the early stages of the company [20].

5.2 Team mates – capabilities through growth

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

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

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

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

A software startup may end up to a situation opposite to the above if the founder is not competent enough to foresee the future challenges realistically and to evaluate the software development skills of the applicants she is going to hire. A crude mistake in that may lead to a total failure in developing the product or service, laying-off the first team, gathering a new team and starting the work from the beginning if the resources allow [21]. If the founder ends up to such a situation, the most crucial step is to find an experienced software professional, carry out reasonable introduction to her, ensure her commitment towards the startup and let her take care of gathering a new team [21].

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

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

5.3 Piloting the implementations – capabilities through learning

The recent approaches of product and software development, incremental, agile and lean development practices, and developments in both hardware and software platforms

have lowered the technology-related entry barriers of software startups. The principles of the lean startup [9], crystallize an iterative, incremental development of minimum viable products (MVPs) that are functional solutions providing the key functionality and characteristic of the targeted product with a minimum development effort. The feasibility of MVPs is measured with the actual customers, fitting well to software startups, developing new, innovative products or services in a small team.

A development process following the principles of the lean startup provides a software startup with several benefits that help the startup to tackle the challenges listed in tables 1 and 2. Though its key topic is avoiding wasted efforts on products or services without real business potentials,

1. iterative and incremental development is easier for small teams,
2. iterative and incremental development creates faster tangible results that provide the team with faster learning, and
3. close customer co-operation provides the team with right learning, guiding the team faster to right directions.

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

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

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

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

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

The capability increments gained by learning during the iterative and incremental development process are, in turn, problem specific, company specific and customer specific. Thus, it is reasonable to claim that such capabilities are more valuable for the startup than the ones from the past. The capabilities acquired in the actual product development context provide a software startup with a resource base that is rare, difficult to imitate, or difficult to substitute, which in terms of the resource-based-view gives the startup a sustainable competitive advantage [22].

5.4 Capability structure of the initial team

Out of the findings we could figure out a capability structure of a software startup's initial team that combines the capability areas, the team roles and the means of capability acquiring identified in our studies. The structure is presented in figure 1 [21].

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

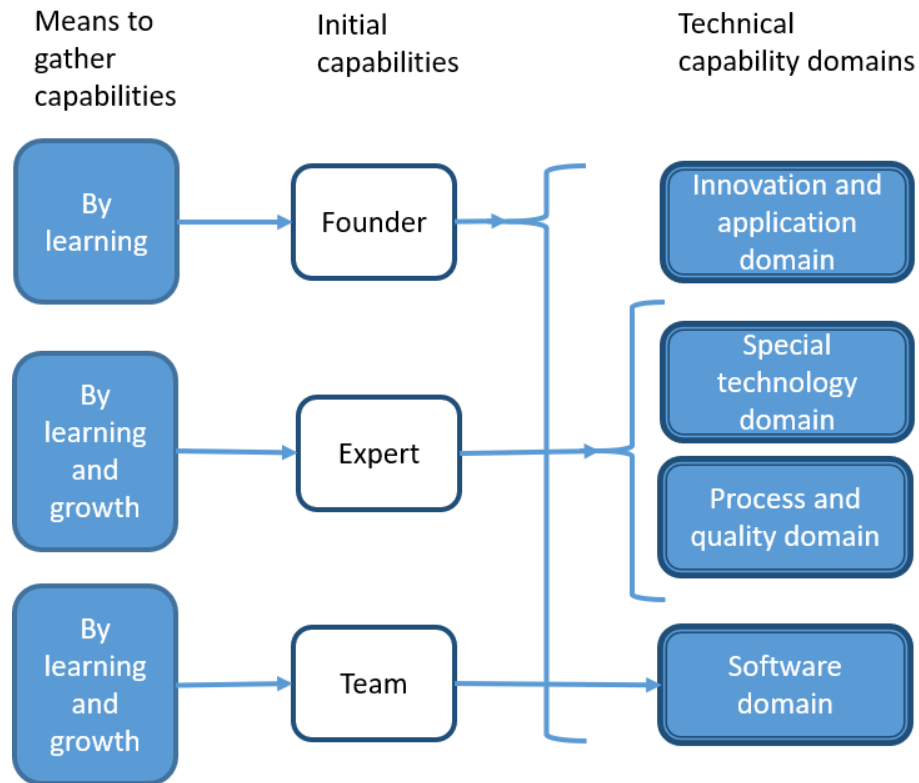


Figure 1. Capability structure of a software startup's initial team

6 Conclusions

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

In a capable software startup there must be an innovation owner, a person with confidence to the innovation and willingness to bring it further to a product. From our study group's perspective that seems to be self-evident, because all founders were committed to the innovation and to the future product. It is known, however, that this is not always the case, but a startup may need to struggle in finding out an idea to bring further [9, 10]. From a just-graduated student's point of view a committed idea owner is of crucial importance – one should avoid employments in software startups without reasonably strong commitment of bringing the innovation to a product. Even though the founder's or the team's commitment does not guarantee any business success for

the developed product, it provides a new comer with a clearer focus and a more stable directions to the product development. That, in turn, offers her better learning points how the development work in software startups is and how she could utilize those learnings in the future career.

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

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

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

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

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

References

1. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: A systematic mapping study. *Information and Software Technology*. 56, 1200–1218 (2014).
2. Bosma, N., Van Praag, M., Thurik, R., De Wit, G.: The Value of Human and Social Capital Investments for the Business Performance of Startups. *Small Business Economics*. 23, 227–236 (2004).

3. Marvel, M.R., Lumpkin, G.T.: Technology entrepreneurs' human capital and its effects on innovation radicalness. *Entrepreneurship Theory and Practice*. 31, 807–828 (2007).
4. Lazear, E.P.: Balanced Skills and Entrepreneurship. *The American Economic Review*. 94, 208–211 (2004).
5. Blank, S.: *The four steps to the epiphany: Successful strategies for startups that win*. K&S Ranch Press (2013).
6. 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, 585–604 (2016).
7. Giardino, C., Unterkalmsteiner, M., Paternoster, N., Gorschek, T., Abrahamsson, P.: What do we know about software development in startups? *Software, IEEE*. 31, 28–32 (2014).
8. Klotins, E., Unterkalmsteiner, M., Gorschek, T.: *Software Engineering Knowledge Areas in Startup Companies: A Mapping Study*. Presented at the (2015).
9. Ries, E.: *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Crown Publishing Group (2011).
10. Bosch, J., Olsson Holmström, H., Björk, J., Ljungblad, J.: The Early Stage Software Startup Development Model: A Framework for Operationalizing Lean Principles in Software Startups. *Lean Enterprise Software and Systems*. 1–15 (2013).
11. Ojala, A.: Business models and opportunity creation: How IT entrepreneurs create and develop business models under uncertainty. *Information Systems Journal*. 26, 451–476 (2016).
12. Alvarez, S.A., Barney, J.B.: Discovery And Creation: Alternative Theories of Entrepreneurial Action. *Strategic Entrepreneurship*. 26, 11–26 (2007).
13. Seppänen, P., Liukkunen, K., Oivo, M.: Little Big Team: Acquiring Human Capital in Software Startups. In: *International Conference on Product-Focused Software Process Improvement*. pp. 280–296 (2017).
14. Seppänen, P., Oivo, M., Liukkunen, K.: The initial team of a software startup, Narrow-shouldered innovation and broad-shouldered implementation. In: *To be published in 22nd ICE/IEEE International Technology Management Conference* (2016).
15. Lethbridge, T.C., Sim, S.E., Singer, J.: Studying software engineers: Data collection techniques for software field studies. *Empirical software engineering*. 10, 311–341 (2005).
16. Marshall, M.N.: The key informant technique. *Family Practice*. 13, 92–97 (1996).
17. Cruzes, D.S., Dybå, T., Runeson, P., Höst, M.: Case studies synthesis: a thematic, cross-case, and narrative synthesis worked example. *Empirical Software Engineering*. 20, 1634–1665 (2015).
18. Cruzes, D.S., Dybå, T.: Recommended steps for thematic synthesis in software

- engineering. In: International Symposium on Empirical Software Engineering and Measurement. pp. 275–284 (2011).
19. Seppänen, P., Tripathi, N., Oivo, M., Liukkunen, K.: How Are Product Ideas Validated? In: International Conference of Software Business. pp. 3–17 (2017).
 20. Nguyen-Duc, A., Seppänen, P., Abrahamsson, P.: Hunter-gatherer cycle: a conceptual model of the evolution of software startups. In: Proceedings of the 2015 International Conference on Software and System Process. pp. 199–203. ACM (2015).
 21. Seppänen, P.: Balanced initial teams in early-stage software startups : building a team fitting to the problems and challenges. (Doctoral Dissertation, University of Oulu, Finland) (2018).
 22. Barney, J.B.: Firm Resources and Sustained Competitive Advantage. *Journal of Management*. 17, 99–120 (1991).
 23. Duc, A. N., & Abrahamsson, P.: Exploring the Outsourcing Relationship in Software Startups: A Multiple Case Study. *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, 134–143 (2017).