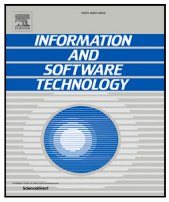
Information and Software Technology 160 (2023) 107224

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof



StartCards — A method for early-stage software startups

Kai-Kristian Kemell [a,*], Anh Nguyen-Duc [b], Mari Suoranta [c], Pekka Abrahamsson [d]

[a] Department of Computer Science, University of Helsinki, Yliopistonkatu 3, Helsinki, 00014, Finland
[b] Business School, University of South Eastern Norway, Gullbringvegen 36, Bø i Telemark, 3800, Norway
[c] Jyvaskyla University School of Business and Economics, University of Jyvaskyla, Seminaarinkatu 15, Jyvaskyla, 40014, Finland
[d] Faculty of Information Technology and Communication Sciences, Tampere University, Kanslerinrinne 1, Tampere, 33014, Finland

ARTICLE INFO

Keywords:
Software startups
Requirements engineering
Validation
Software engineering method
Action research

ABSTRACT

Context: Software startups are important drivers of economy on a global scale, and have become associated with innovation and high growth. However, the overwhelming majority of startups ends in failure. Many of these startup failures ultimately stem from software engineering issues, and requirements engineering (RE) ones in particular. Despite the emphasis placed on the importance of RE activities in the startup context, many startups continue to develop software without a clear market or customer, having never had meaningful contact with their would-be customer.

Objective: We develop a method aimed at early-stage startups that is intended to help startups through the initial stages of the startup process: StartCards. The method emphasizes the importance of idea and product validation activities in particular in order to tackle anti-patterns related to (a lack of) RE in startups. This method is based on existing literature, both grey and academic literature.

Method: StartCards was developed using the Canonical Action Research (CAR) approach, over the course of 4 AR cycles. During the AR process, the method was used by 44 student startup teams in a practical course setting. Data from the use of the method was collected through self-reporting in the form of modified learning diaries, mentoring meetings with the startup teams, and a qualitative survey.

Results: We consider the current version of StartCards useful for early-stage startups based on the data we have collected. The method can also be used as a pedagogical tool in startup education.

Conclusions: The paper presents the first published version of the method. While work on the method continues, the method is deemed ready for use.

1. Introduction

Software startups remain important drivers of economy across the globe. There are currently more than 140 000 startups in Europe, and roughly a third of these have managed to acquire at least one round of funding [1]. In total, this adds up to some €43,3 billion invested in European tech startups in 2019 [1], with similar projected numbers for 2020 despite the unforeseen effects of the pandemic year. US startups, on the other hand, saw investment up to 140 billion $USD in 2019 [2]. Yet most software startups fail [3–5], and up to 98% of all new product ideas in general fail [6] (p. 3). As a result, most of these investments are wasted.

Various extant studies (e.g., [5,7,8]) investigate the reasons behind these software startup failures and the challenges startups face. Many of these studies suggest that software startups struggle with Software Engineering (SE) as much as they struggle with business, and sometimes these two aspects are very interlinked in the software startup context. Klotins et al. [8] in particular argue that many of the seemingly business-related failures may in fact stem from SE factors, and especially requirements engineering.

While there is arguably no silver bullet for business or software startup success, it seems that experienced startup founders are more likely to be successful in their business ventures [9]. As such, there seems to be something that can be gained from various lessons learned and good practices in the startup context as well. To this end, various startup practices are discussed in academic literature and grey literature alike, including the Minimum Viable Product (MVP) [10] [11], pivoting [12,13], and the build–measure–learn loop [14].

On the other hand, the suitability of existing SE practices and methods has been questioned in the startup context, as software startups differ from more traditional software organizations, including small and micro-sized companies, in various ways [15]. Startups are temporary

* Corresponding author.
E-mail address: kai-kristian.kemell@helsinki.fi (K.-K. Kemell).

https://doi.org/10.1016/j.infsof.2023.107224
Received 4 February 2022; Received in revised form 16 March 2023; Accepted 3 April 2023
Available online 9 April 2023

organizations. They either grow into established software organizations or fail somewhere along the way. Startups are characterized by disadvantage [16], which can stem from various factors that can vary from startup to startup, including uncertainty, lack of resources, inexperienced team, and time pressure [17,18].

These differences between startups and other software organizations are also a relevant issue from the point of view of software development processes. When it comes to SE, startups seem to seldom utilize methods at all and prefer singular agile practices [17], especially early on in the startup life cycle [19]. It is not clear why this is the case. One explanation, as Bosch et al. [20] argue, could be that: "[Agile methods] are mainly applied in situations where the problem is fairly well understood but the solution is not. In a startup context, however, neither the problem nor the solution is well understood".

In this light, methods that account for the unique context of software startups could help. However, few such methods exist. The lean startup of Eric Ries [14] is the most well-known such method. In practice, though, the lean startup has been difficult to utilize, and is more of a philosophy than a method, consisting of a few actionable practices but no process. As a result, it has been difficult to utilize in practice [20], and various handbooks and guidelines (e.g., [21]) to help software startups and other organizations use it have been published.

Many of the challenges faced by startups, as highlighted by Wang et al. [7] and Klotins et al. [8] are related to Requirements Engineering (RE). In startup literature, these activities are typically referred to as 'validation' activities, and focus on validating the idea and the solution (software). I.e., validating whether the problem the startup is trying to solve is a real problem (or whether the value they are trying to deliver has any real value to their would-be customers or users), and whether the product or service they have chosen to do so with is the right choice for the given business context [7].

Given the lack of suitable existing methods for startups, we seek to address this gap by developing one, placing emphasis on validation activities. In this paper, we propose a method for early-stage software startups: StartCards. By *early-stage startup*, we refer to startups still working on their product or service concept and those in the early stages of the development process (as we discuss in more detail in Section 3.1). The method focuses on helping such startups validate as early as possible whether their idea is worth pursuing in the first place, as working on an idea without validating it by involving customers is dangerous [8]. This is line with the idea of failing fast, which is emphasized in lean startup [14]. Of course, the idea is to *not* fail, but if the idea was never going to work in the first place, it is best to realize this early on, which is why we have chosen to focus on early-stage startups in particular. As Klotins et al. [22] summarize: "exploring the problem domain and user needs is one of the key practices in early stage start-ups".

If problems with the business idea are discovered while validating it, a startup can then either pivot [23], i.e., change the idea in some way, or re-evaluate its plans entirely. While validation is, ideally, a continuous process where multiple Minimum Viable Products (MVPs) [10] of different types are built, starting with low fidelity MVPs early on and utilizing increasingly technical prototypes later on, this process is best started early. By already validating early on, it is possible to avoid spending resources working on an unfeasible idea, or to confirm early on that the idea is indeed worth pursuing. Larger pivots (such as drastically changing the underlying business idea itself, as opposed to, e.g., simply targeting a different customer or user segment) can be easier to make earlier on when there are less sunk costs involved, and especially if the product or service is still just an idea or only in the early stages of the development process.

The method we propose in this paper, StartCards, contains a set of good practices for startups, which is based on existing academic literature and grey literature. These practices are depicted as a deck of cards. Though its primary focus is on validation activities, the method contains other good startup practices as well. The method has been developed using an Action Research (AR) approach, and specifically the canonical AR approach described by Davison et al. [24], based on the cyclical AR model of Susman & Evered [25]. Using this approach, the method has been developed iteratively over the course of four years. In the process, we have used data from 44 startups while deploying the method in a learning "through" entrepreneurship [26] practical course setting.

The rest of this paper is structured as follows. In Section 2, we discuss the theoretical background of this paper. In Section 3, we present the method this paper proposes. In Section 4, we discuss the research method of the paper. In Section 5, we present the results of the AR process. In Section 6, we discuss the implications of the method and the results of the AR process, as well as validity threats. Finally, Section 7 concludes the paper.

2. Background

The theoretical background of this paper is split into two subsections. In the first one, we discuss SE in startups. In the second subsection, we discuss requirements engineering and idea and solution validation.

2.1. Software engineering in startups

As touched upon in the introduction, one of the main arguments behind software startup research in SE has been their uniqueness: "software startups are quite distinct from traditional mature software companies, but also from micro-, small-, and medium-sized enterprises, introducing new challenges relevant for software engineering research". [15]. Wang & Nandhakumar [16] summarize that startups are characterized by disadvantage. This disadvantage, extant literature argues, stems from various factors, such as some of the following characteristics commonly associated with software startups: (1) highly reactive, (2) innovation, (3) uncertainty, (4) rapidly evolving, (5) time-pressure, (6) third party dependency, (7) small team, (8) one product, (9) low-experienced team, (10) new company, (11) flat organization, (12) highly risky, (13) not self-sustained, (14) lack of resources, and (15) little working history [17].

Because of this unique context, existing research findings, lessons learned, and SE methods aimed at traditional software organizations may be poorly applicable to software startups. According to Paternoster et al. [17] "agile and more traditional methodologies struggle to get adopted by startups due to an excessive amount of uncertainty and high time-pressure" and "software development practices are reported to be adopted only partially and mostly in a late stage of the startup life cycle". In this regard, however, an interesting question to pose is whether startups even *should* be using agile methods at all, if, as Bosch et al. [20] argue, they are not so well-suited for startups.

In SE and IS in general, we consider method use ideal and it is the norm out on the field as well. Not using methods goes against our idea of conventional SE. While we now know that startups are seemingly averse to utilizing agile methods and other traditional SE methods [17], we do not know exactly *why* this is the case. Perhaps, indeed, these methods are not so suitable for startups, and hence go unused. Yet startups do continue to utilize individual agile practices [19], although often in an ad hoc fashion [27], and the iterative way of working at the heart of agile is present in lean startup [14] as well. Not all existing practices, as such, seem completely unsuited for the startup context. However, the agile practices startups favor [28] seem to differ from those commonly favored in the industry in general (based on, e.g., the State of Agile Report [29]).

To characterize how startups develop software, Giardino et al. [18] propose the Greenfield Startup Model. According to this model, software development in startups is characterized by a severe lack of resources, which is a characteristic widely discussed in other extant literature as well (e.g., [7,30]). According to the model, this lack of

resources leads to the team being the one (and perhaps only) key resource the startup has (as also emphasized by, e.g., [31–33]), as well as quality having low priority (as also discussed by, e.g., [22]). Together, these result in speed being the focus in product development, which in turn results in technical debt being accumulated. In the Greenfield Startup Model, initial growth later hinders performance, as a result of said technical debt [18].

Technical debt is "a metaphor for immature, incomplete, or inadequate artifacts in the software development life cycle that cause higher costs and lower quality in the long run". [34]. In software startups, product quality is seldom a priority [8] and speed is prioritized and corners are cut to develop faster, leading to the accumulation of technical debt [18]. This is arguably useful, as it facilitates the lean startup [14] idea of failing fast by shortening the time it takes to develop the system. Should the startup fail, the technical debt never realizes in the first place, and even if the startup does not fail, being faster has its benefits when dealing with a lack of resources. As to how this technical debt is handled when startups do not fail, it seems to be common for startups to later simply abandon these systems or components riddled with technical debt in favor of new ones, as opposed to attempting to refactor or otherwise fix them [19].

Startups also commonly focus on one product or service [17,30]. Consequently, the entire company is built around that one development endeavor, making business aspects closely intertwined with SE. This ties to the argument of Klotins et al. [8] who argue that some of the seemingly business-related failures in startups may in fact stem from SE decisions, as we discussed in the introduction.

We have established that startups seldom utilize conventional SE methods. However, SE methods specifically tailored towards startups are scarce. The most common method associated with startups is the lean startup methodology of Ries [14]. However, though it is often referred to as a methodology, the lean startup is more akin to a philosophy featuring a handful of practices as opposed to a process. Just as "doing agile" can mean many things [35], "doing lean startup" can mean many things.

On the other hand, the practices included in lean startup have become widely discussed in startup literature. The main practices discussed in lean startup are the MVP, pivoting, and the build–measure–learn loop [14]. While the extent of their utilization out on the field remains a question, they are also widely discussed in other grey literature on startups.

Finally, existing studies have presented some tools aimed at software startups. Bosch et al. [20] present a model to help startups operationalize lean startup, and specifically its Build–Measure–Learn (BML) loop. Melegati et al. [36] propose HyMap, a tool to help startup practitioners validate business assumptions by turning them into testable hypotheses. Such tools are useful for what they are tailored towards. However, we are not aware of a method being proposed by extant literature. With the lack of method use in startups being considered an issue, we aim to tackle this problem by presenting one aimed at startups in this paper (see Section 3). The method focuses on validation activities in particular, which can be likened to requirements engineering in conventional SE, as we discuss next.

2.2. Requirements engineering and idea and product validation

Zave [37] defines Requirements Engineering (RE) as follows: "Requirements engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families". More concisely, it could be said that requirements engineering is about understanding 'what' should be built and 'why' [38]. This is achieved by understanding the needs of the stakeholders, such as users or clients, and documenting and analyzing them to formulate requirements for the system being built [38,39]. Various practices for doing so exist, including face-to-face communication, customer involvement, and prototyping, among various others (as, e.g., discussed by Inayat et al. [39]).

Startups also carry out RE [40], especially past the earlier stages. Once the startup has secured some initial customers or would-be customers, it is possible to work with them using more conventional RE approaches, as discussed by some of the case startups of Melegati et al. [40]. For early-stage startups, on the other hand, it can be difficult to utilize these conventional RE practices. Many of these practices rely on interacting with users or customers [39], and startups are known to develop software without having had meaningful contact with their planned users or customers [8]. Consulting a customer to formulate requirements in a traditional manner cannot be done without a customer to consult.

Potential users should still be involved in order to understand their needs, however [8], but doing so requires different practices. Such practices in startup literature are commonly referred to as *validation* activities as opposed to requirements engineering ones, even though their goal is ultimately the same: to understand the 'what' and 'why' of the planned (or on-going) development endeavor. In the context of conventional RE, validation has a different meaning. There, it refers to the last stage of the RE process, where the final set of requirements before implementation by having an analyst check them [41,42]. For early-stage startups still in the process of conceptualizing their business idea, understanding *who* their (future) users are or should be is also important (e.g., as denoted by the business model canvas [43]). From this follows that, in addition to the 'what' and 'why', 'for whom' is also a question an early-stage startup may need to pose.

As a result of early-stage startups having no clients to consult (and sometimes not even having a clear idea of their target users at all other than in the form of an assumption), the context of RE in startups can be very different from conventional RE, and is often referred to as validation as opposed to RE in startup literature. As a result of this unique context, RE or validation practices aimed at the startup context have been proposed. In particular, all the key practices of lean startup [14] (i.e., the Minimum Viable Product (MVP), pivoting, and the Build–Measure–Learn loop) focus on validation. According to lean startup, every startup should build MVPs in order to validate their business idea [14]. To this end, Ries [14] posits that MVPs are data collection tools that should only be built with some clear goal in mind for said data. In practice, an MVP be anything from a landing page to a mock-up or a functional prototype. While a prototype can be an MVP, various other types of MVPs exist as well [10,11,14].

In lean startup [14], MVPs are a part of the build–measure–learn (BML) loop that recommends an iterative working approach that utilizes MVPs. In the BML loop, as its name indicates, an MVP is first built, then used to measure something while data is collected in the process, and that data is then used to learn something about the business idea or the product being built. In this fashion, MVPs should be used as a part of an iterative process where multiple, different types of MVPs provide validation for the idea (though startups seem to seldom do so in practice [10]). Finally, based on the data gathered in the process, a startup should evaluate whether to pivot, i.e., change direction, or to persevere, i.e., keep executing the current plan [14]. Pivots can vary in scope, from changing the entire business idea to changing the platform the software is being developed for, or changing the customer segment being targeted [12,14,23]. However, though MVPs are intended to be used as data collection tools, their use in practice is more multi-faceted. MVPs are often reused and retooled where possible [10].

Finally, though both academic and grey literature places emphasis on these types of validation activities, lack of validation remains an issue in startups [8,36]. This can result in a situation where software is developed without a solid understanding of the core value its users want it to provide [8].

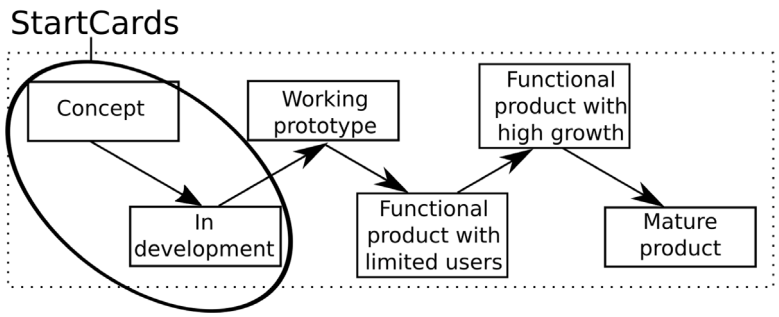


Fig. 1. Our method's scope in the Startup life cycle.
Source: Wang et al. [7].

3. StartCards

This section presents the method itself: StartCards. In the first subsection, we discuss the theoretical background of the method. In the second subsection, we present the method. In the third subsection, we discuss how the method is used in practice.

3.1. Theoretical background of the method

The method of this paper is intended to help early-stage software startups. The focus of the method is on (1) formulating the business idea and the product idea, and (2) validating the idea. We have chosen to focus on early-stage startups because developing software without conducting proper validation continues to be an issue in software startups [8,36], as we have also discussed in the preceding sections. We hope to discourage such situations by emphasizing the importance of validation early on.

When speaking of *early-stage* startups, we refer to a startup life cycle model presented in existing literature to illustrate what this means. In this regard, we use the life cycle model of Wang et al. [7] to position the method in Fig. 1. In this model, the StartCards are aimed at startups in their earlier stages who are still working on the concept and its early, initial development, as illustrated by the circled area labeled *StartCards* in Fig. 1. In the context of another life cycle model we have proposed in an existing paper [44], the method is aimed at the first stage, the *pre-startup stage*.

As we began to design the method in practice, we chose a card-based approach. Using cards in SE to facilitate work is common practice. One example of such use of cards is Planning Poker or Scrum Poker, which is among the more commonly utilized practices according to the 14th State of Agile report [29]. User stories are also a card-based practice.

The cards describe various startup practices based on existing literature, both academic and grey literature. The cards themselves also include references to literature, intended to serve as further reading for the users of the cards. For this purpose, we have utilized literature we felt could be of interest for practitioners. These references are detailed in Table 1.

The cards describe practices. However, these are rather general practices, and many of the cards could arguably be split into more atomic practices. E.g., there are various types of MVPs that could each have a card dedicated to them, as opposed to having one MVP card supposedly covering them all. However, this is not done because (1) we wanted to keep the method concise and lightweight, and (2) the references on the cards are intended to tackle this issue by providing further reading that can help the users of the cards familiarize themselves with more specific practices in the context of each card. The suitability of such more atomic practices can also vary greatly between startups. This is nonetheless a potential validity threat related to construct validity that we discuss in the validity threats section. Describing practices is a recurring topic of discussion in SE [45].

3.2. Method description

StartCards, as the name implies, is a card-based method that consists of a deck of 14 cards. Each card describes a startup practice. These practices, and consequently the cards, are based on both academic and practitioner literature on software startups. The cards are numbered and sorted to reflect startup life cycle models, in particular that of Wang et al. [7]. Table 1 provides an overview of the cards, listing the titles of the cards and providing brief description of each card. The method can be found on figshare: https://doi.org/10.6084/m9.figshare.20151722.v1.

The content on each card is split into three parts: (1) motivation (why this is important), (2) what to do (the activity itself), and (3) common mistakes (to avoid), i.e., the "don'ts". Additionally, the cards include references to literature as further reading. For example, the references on the MVP card discuss different types of MVPs in more detail.

The cards can be utilized in conjunction with existing methods. In fact, the cards actively refer to other tools and practices related to the contents of the cards, and encourage their users to actively reflect on their way-of-working. The cards also support and encourage iterative work. Iterative development is the norm in agile and SE in general today, and startups in particular are also encouraged to work iteratively from a business point of view. For example, the build–measure–learn loop of the lean startup is entirely based on working in an iterative fashion.

3.3. How to use the method in practice

As StartCards consists of a deck of cards, using the method is about using the cards. While the cards do not form a strictly linear process, the cards are numbered and should initially be used starting from the first one and proceeding in order. The cards start with early ideation and proceed towards solution validation towards the end of the deck. Activities on earlier cards may have to be carried out before activities on later cards can be carried out. For example, in order to validate an idea, one has to first formulate the idea. However, this is not a linear process in practice. For example, a pivot may result in taking steps backwards should one treat it as one, and similarly, the use of MVPs is intended to be an iterative process.

To start using the cards, we recommend that you start from card 1 and work your way up. For each card, consider whether the topic is (still, or currently) relevant for your startup. This is best done by involving multiple startup team members. If you feel that you have already passed the point where, e.g., the Appealing Idea (1) card is relevant, feel free to disregard it. Some practices are more relevant for very early-stage startup than others. Overall, the deck is primarily aimed at early-stage startups still working on the initial version of their product.

The cards are modular. They encourage their users to pick the cards that are most relevant for their current situation. For example, if you

Table 1
The cards included in the method.

#	Card title	Description	References
1	Appealing Idea	Advice for idea generation.	[4,14,46]
2	Great Pitch	Advice for presenting the idea briefly (i.e. "pitching").	[47,48]
3	Validating the Appealing Idea	Advice for idea validation.	[4,14,21,36]
4	Get the Right Team Together	Emphasizes the importance of the startup team.	[32,33]
5	Create a Business Model	Advice for creating a business model.	[33,43]
6	Mapping the Competition	Advice for understanding the competition in the target market.	[21]
7	Establish Your Initial Way-of-Working	Tips on establishing initial wok processes.	[49,50]
8	Validating the Potential Solution	Advice for solution (product/service) validation.	[10,20,36]
9	Frequent Early Pivots	Emphasizes the importance of pivoting (changing direction) when the idea of some part of it starts looking unviable.	[12,13,23]
10	Utilize Metrics	Advice for utilizing data in the form of metrics in various ways.	[51]
11	Minimum Viable Product	Advice for using MVPs to validate the idea and solution.	[4,10,11]
12	Startup Spirit	Emphasizes the importance of having the mindset of an entrepreneur.	–
13	The Learn–Measure–Build Loop	Further advice for using MVPs in a planned manner.	[4,14,20]
14	Calculate the Financial Metrics	Advice on how to better convince potential investors with financial numbers.	[47,52]

wish to build an MVP, cards 3, 8, 10, 11, and 13 are all related to MVPs in some way. In this fashion, the cards can be arranged into various smaller processes as they are used.

The cards are meant to provide motivation and instructions for utilizing the practices. However, they ultimately only contain the basic idea of the practices. For example, the cards contain the basics of creating and utilizing MVPs, but for tips on how to best utilize specific types of MVPs, further reading is encouraged. The aim of the cards is to encourage their users to seek more information on the topics if and when needed, after providing an introduction to the practices. The Internet is full of various tips and tricks on various startup matters, including the ones described in the cards. As each startup and business idea is unique, one cannot recommend any one practice that would be the best for validating one's business idea in every case, for example.

4. Research method

StartCards were devised using Action Research (AR), over the course of 4 AR cycles and 4 years. More specifically, we utilized the Canonical Action Research (CAR) approach described by Davison et al. [24]. CAR provided us a way of developing the method iteratively, as we wanted to keep deploying it in practice to gather feedback, then changing it based on the feedback where applicable, and deploying it again. While AR is a common methodology in organizational research and we deployed the method in software startups, we did so in an educational setting in a project-based course.

The rest of this section details this process as follows. In Section 4.1 we discuss the study setting (the course) in detail. In Section 4.2 we discuss the types of data used in this paper (data collection). In Section 4.3 we discuss how this data was analyzed. In Section 4.4, we discuss the cyclical AR process, going over the five stages of CAR and its five principles [24] in the context of this study.

4.1. Study setting

StartCards were developed using data from a course on software startup entrepreneurship, *Venture Lab & Lean Startups*, at the University of Jyväskylä (JYU) in Finland. This course was originally taught in the Faculty of IT, but became a joint course between the IT Faculty and the Business School during the AR process. The course was a master's level course aimed at both IT (Information Systems and SE) and business students. Thus, students taking the course were typically at least third year students. In the curriculum of the IT faculty, the course was a part of the startup entrepreneurship study module. Positioning this course in existing typologies for such courses, this course was a learning "through" entrepreneurship course [26] or an action-based entrepreneurship course [53].

During the course, students created startups as teams based on ideas they came up with at the start of the course or based on existing ideas they already had in mind. Additionally, students already involved in startups were encouraged to work on that startup in the context of the course by recruiting some course participants to work with them for the duration of the course. Some of these startups have participated in some of our extant studies as well (e.g., some of the cases in [33,50] were 'real' startups from this course). Thus, the startups in the course could be categorized into three types: (1) pre-existing, real-world startups, (2) startups founded at the start of the course that were intended to become real businesses, and (3) startups that were purely course projects. In some cases, type (2) startups could become type (3) startups during the course if the team concluded, as a result of the practical course activities, that the idea did not seem feasible after all.

At the start of the course, the students were asked to think of potential startup ideas for the first lecture. During the first lecture, these ideas were then pitched to the class. While organizing into teams, the students could team up with a student whose idea they found particularly interesting, if they did not have an idea of their own

Table 2
Generalized course outline.

Week	Lecture topic(s)	Weekly task(s)
1	Business ideas, forming teams	Forming a team. Deciding on business idea. Preparing an initial pitch and pitch materials for the idea.
2	Business models	Using business model canvas to describe business. Idea validation by collecting secondary and primary data to support the idea. Mapping the current competition on the target market.
3	Startup methodologies and tools	Further idea validation by collecting secondary and primary data to support the idea. Mapping the current competition on the target market.
4	Customer development	Planning an MVP, building it, and using it to validate idea/solution.
5	Startup fundamentals, IPRs, metrics	Devising a plan for using different metrics and executing it to what extent currently possible.
6	Acquiring funding	Preparing detailed financial calculations (expenses vs. revenue) for the idea, while being advised by a practitioner expert.
7	Pitching	Preparing a high-quality final pitch.
8	Ending event	Idea is pitched at a live event for a panel of practitioner experts.

they would have rather worked on. Similarly, those students with pre-existing startups could also pitch their business at this stage. Some re-organizing was done to create teams of 4–5 students, in order to avoid pairs or teams of ten students. By having at least 4 students in a team, one student per team could drop out during the course without reducing their former team into a pair of students.

The course was punctuated by weekly lectures and weekly mentor meetings. The lectures were on Mondays, while the mentor meetings were set up based on the schedules of the mentors and the teams. These mentor meetings were conducted by authors 1, 3, and 4, in addition to a research assistant who acted as a fourth mentor. The intended duration of the mentor meetings was 20 min per team per week. During the weekly mentor meetings, the teams received guidance and were coached by the teaching team (and occasionally by practitioner experts) based on their current situation and progress.

During the rest of the week, the teams were to work on their startups. Each week there was a set of tasks to be completed by each team, which set the bar for the minimum amount of work for each week. In addition to these mandatory tasks, the teams were encouraged to work on their ideas as they best saw fit in order to progress. Each week, during the lecture, the teams showcased their progress by means of a pitch, or by presenting their weekly progress in some other manner (this varied between course iterations). The course proceeded in this fashion for 8 weeks.

The contents of the course are outlined in Table 2. In the table, the theme for each week is outlined. However, this is a generalization as we discuss the use of the method over the course of four course instances (2018, 2019, 2020, and 2021), and the schedule was not the exact same for each year. While the general outline is largely the same between all iterations of the course, some lectures may have taken place a week earlier or later due to guest speaker availability.

Finally, while the course was technically a linear process, we acknowledge the iterative nature of working on a startup. Pivots were expected and encouraged during the course. To facilitate this, the weekly tasks could be completed regardless of the stage the startup was at. For example, when building an MVP, the team could choose an MVP that best matched their current progress and idea. After a pivot, the teams were also urged to repeat some tasks for the new idea, such as carrying out validation activities for the new idea as well.

4.1.1. The role of the method in the course setting

During the course, every week the students were introduced to one or more cards from the StartCards card deck. These cards were distributed at the end of each lecture. Each team received a physical copy of the card(s) before leaving the classroom in the 2018 and 2019 iteration of the course. Due to COVID-19, the 2020 iteration was entirely online and as such no physical cards were handed out. The 2021 course was carried out physically with online participation being possible, although the cards were once again digital as a result to accommodate this hybrid more of teaching.

The use of the cards and the method in general was voluntary. The students were directly told that their use or lack of use of the method would not affect grading in any way. Similarly, they were told that the method is under development and that if they used the cards, any feedback, positive or negative, would be welcome, and, again, would have no bearing on grading. Documenting their use of the cards and reflecting on it was simply one way for the students to produce content for their learning diary during the course (which we discuss next in Section 4.2). By introducing the method in this fashion, we wanted to see whether the method was considered useful. Given that its use was voluntary, we hoped that the students would only use it if they considered it to be beneficial to their startup in some way.

4.2. Data collection

We utilized multiple (three) types of data during the AR process, as is typical in AR. Data was collected from 44 startup teams in total, as depicted in Table 3. Table 4 describes the types of data collected during each of the four AR cycles, and Fig. 2 describes when, during the course iterations, this data was collected. First, we collected data through modified learning diaries we call Startup Scratch Books (SSB). Secondly, we utilized data from the weekly mentor meetings. Thirdly, at the end of the fourth and final AR cycle, we collected survey data from the participants of the 2021 course iteration.

First, the *SSBs*. We have discussed the concept of the SSB in detail in an existing paper [54] (and it has also been featured in a teaching innovation competition [55]). The SSB is a novel concept that, in terms of existing approaches, can be best likened to a learning diary. The aim of the SSB is to "open the startup black box". We know that startups work largely in an unorganized or ad hoc fashion, but understanding their work approaches in more depth is still of interest in SE. There

Table 3
Startup teams involved in the study.

Year	Startup teams
2018	11
2019	8
2020	12
2021	13
Total	44

Table 4
Data types used in the study.

Data type	Description	AR Cycle
Startup Scratch Books (SSBs)	The SSBs can be considered modified learning diaries. Each year, each team produces an SSB during the course. The SSBs were analyzed for any content related to the method and its use.	1–4
Mentor meetings	Each year during the course, every team had a weekly mentor meeting with a member of the teaching staff (or a practitioner expert). During these mentor meetings, the use of the method was occasionally discussed. These can be likened to informal interviews.	1–4
Survey	A post-course survey was conducted to collect additional data about the use of the method during the 2021 course.	4

is seldom any paper trail to be analyzed for startups. A failed early-stage startup simply disappears in many cases, especially if the failure happens before the startup even becomes a company in the legal sense. As we teach a course on startups where students work on startups, we wanted to better understand what happens inside these early-stage startups during our course. We wanted to also look at their inner workings as opposed to simply looking at their results in the form of pitches, or as they discuss them in the mentor meetings.

The SSB is not just a learning diary, but also a scrap book. In addition to traditional learning journal (or learning diary) [56] content such as reflection, the SSBs also include work products such as sketches, data, and even communication logs if the team feels comfortable including them. In this fashion, rather than making the students always tell us retroactively what they were doing and why, we wanted them to also *show* us what they were doing.

An SSB proceeds in a chronological order from the start of the course to the end of the course. The content of an SSB is split into logical sections at the team's discretion. The minimum length for an SSB is 100 pages, and while this may sound like a lot on paper, the SSBs are intended to include various full-page illustrations and other such content that ultimately makes reaching that threshold reasonably easy. For this paper, we utilized any content in the SSBs that was related to the use of the cards.

Secondly, the *mentor meetings*. For the purposes of this paper, and from the point of view of AR, these mentor meetings acted as informal interviews, as opposed to formal ones. Data from these meetings related to the method and its use was collected as notes, mental or physical. Only data related to the use of the cards was of interest from the point of view of this paper. The purpose of the mentor meetings overall was to help the startups progress during the course by analyzing their current situation and providing guidance. In the process, we occasionally discussed the method and its use with the teams.

Thirdly, at the end of the 2021 course iteration, we conducted a brief *survey*. In this survey, we asked more directly for feedback on the cards. The two main questions of the survey were (1) "Please describe how the cards were used by you or your project/startup team, if at all" and (2) "Were the cards useful? Why or why not?". Because the method was considered tentatively ready (for potential publication and further testing elsewhere), we wanted to utilize another type of data for triangulation purposes. In an attempt to make untruthful answers less likely, the survey was largely anonymous, with the only identifying data being the team number or team name. Moreover, the survey was conducted after the course had already ended and been graded, in order to avoid any (mis)conception of the responses being able to affect grading negatively.

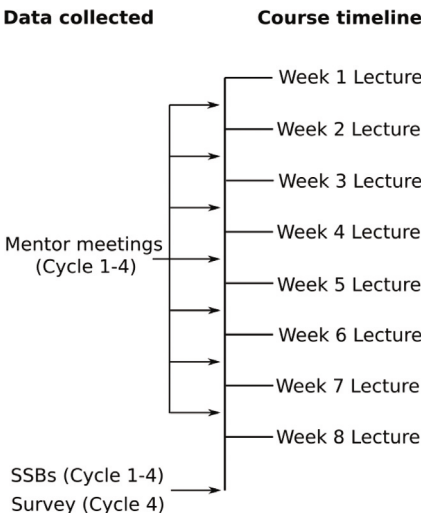


Fig. 2. Data collected during the course.

4.3. Data analysis

As we used three different types of data (Table 4), different approaches to their analysis were also utilized. First, in the case of the SSBs, we went over the contents of each SSB, looking for content related to the method and its use. There were two general types of data in this regard: (a) data documenting the use of StartCards (e.g., mentions of the use of the cards in communication logs, use notes written on the cards themselves, etc.), and (b) reflection and feedback on the use of StartCards.

More specifically, these two types of data related to the use of the method could appear in different forms in the SSBs: (1) as direct feedback about the cards, (2) as reflection about their use, (3) as use notes on the cards themselves, (4) as the inclusion of the cards into the scratch book, or (5) indirectly in the task-related outputs of the team. For example, when carrying out a course task, the use of StartCards could be seen to have impacted the end result even if the use of the cards was not discussed in relation to that task in the SSB. Overall, though, most of the content in the SSBs (which had a minimum length of 100 pages) was simply related to the course in general and was not related to the use of the cards. Arguably, in some cases, it was not straightforward to determine to what extent the method influenced the task if it was not explicitly reflected upon.

In total, data from 44 SSBs across 4 years was analyzed, with each team producing one SSB. In order to provide an overview of the data, we scored the SSBs on a scale of 0 to 3 regarding the use of the method, ranging from no use to high use extent. This scoring is explained in Table 5. However, our reflection was largely based on the qualitative analysis of the SSBs and their other data (e.g., the direct feedback). This use extent analysis was combined with paying attention to *which* of the StartCards were (not) used.

When we discuss our results in Section 5, these SSBs are attributed to numbered teams (e.g., Team 5), but for the sake of anonymizing

Table 5
Framework used for scoring SSBs in Section 5.

Score	Description
0	No data on method use present in SSB.
1	Minimal data on method use. SSB details the use of some cards.
2	Data on method use. SSB details the use of half of the cards.
3	Extensive data on method use. SSB details the use of all or nearly all of the cards.

Table 6
Action research cycles.

AR Cycle	Duration
1	2018
2	2019
3	2020
4	2021

the data, these team numbers are not the same as during the course. Moreover, the startups had company names past the first lecture, which are not included in the data here.

Secondly, data from the *mentor meetings* was collected separately by the authors, with each startup having one mentor per week. Data from these mentor meetings was discussed by the authors in an ad hoc manner as feedback was received. In particular, Authors 1 and 4 regularly sat down to discuss data from mentor meetings that was related to StartCards. Larger changes to the method were done after each cycle or in the diagnosis and action planning stage of a new cycle, but potential future changes were already discussed during the course in this manner. As these mentor meetings were not recorded (given that they mostly focused on the course and the startups rather than the method alone), such discussions and the resulting notes were a way of keeping track of data collected from them.

Finally, the data from the post-course survey of 2021 was analyzed using a qualitative approach. It received one or more responses per team. The responses were binary yes or no responses to whether the method was considered useful, as well as the reasoning behind the response. These responses were used to evaluate what points of improvement the method might still have, in conjunction with the other data.

4.4. Action research process

As mentioned at the start of Section 4, we utilized Canonical Action Research (CAR) [24] in this study. Four AR cycles were conducted between 2018 and the end of 2021, as detailed in Table 6. These cycles were centered around the course discussed in Section 4.1, which took place between October and December of each year. However, work on the method was not limited solely to these time periods.

In the two subsections that follow (Sections 4.4.1 and 4.4.2), we discuss this AR process in detail from the point of view of the five stages and five principles of CAR, as described by Davison et al. [24]. Section 4.4.1 covers the five stages of CAR: Diagnosis, Action Planning, Intervention (Action taking), Evaluation (Assessment), and Reflection (Learning). Section 4.4.2 covers the five principles of CAR: the Principle of Researcher–Client Agreement (RCA), the Principle of the Cyclical Process Model (CPM), the Principle of Theory, the Principle of Change through Action, and the Principle of Learning through Reflection.

4.4.1. AR stages

We provide a brief overview of the AR process in this section. The process is detailed in Section 5 where we report our results.

Diagnosis. At the start of the cycle 1, the diagnosis phase was focused on understanding the problem, i.e., startup challenges and failures, and zoomed in on validation activities as a part of this problem. We studied existing literature, both grey and academic, in order to begin addressing the problem. Moreover, the researchers behind this paper have an extensive list of publications related to software startups (particularly the second and fourth author) and the findings of these studies have contributed to the creation of this method alongside other existing literature.

In cycles 2–4, the diagnosis phases were focused on both better understanding the problem, but also on the method itself and its deployment during the course. We returned to our reflection from the preceding cycle(s) in the diagnosis phases of the later cycles. We also continued to study new startup research at the start of each new cycle so as to ensure that our method was up to date as far as academic research is considered.

Action Planning. In cycle 1, the method was devised during the action planning stage. Its testing was also planned during this stage. We decided to develop the method in the local startup course (Section 4.2). The course had been carried out once in the preceding year of 2017 but in a different fashion. After 2018 it proceeded in the manner described in this paper.

In cycles 2–4, the Action Planning stages of the process were focused on planning the introduction of the method for each course iteration, and also determining whether further changes should be made to the method before the next intervention. This also included plans for data collection. For example, the SSB concept was originally devised in 2018 (cycle 1) and improved during the AR process, primarily between cycle 1 and 2 (see [54] for more details).

Intervention (Action taking). The intervention in all phases has been the introduction of the method into the project context of each team. On a general level, the interventions proceeded in a similar manner in all cycles. However, as we discuss while reporting our results in Section 5, we changed our approach slightly after cycle 2.

Evaluation (Assessment). Evaluation was carried out using the three types of data described in Table 4 in Section 4.2, the analysis of which was discussed in Section 4.3. Evaluation proceeded in a similar manner between all cycles, with the exception of the post-course survey in cycle 4.

Reflection (Learning). The main focus of the reflection phases has been on how to improve the method based on what was learned during the AR cycle. This reflection had two main focuses: (1) were the contents of the cards relevant and useful, and (2) were the cards, and the method in general, being presented in a clear manner. In addition to evaluating the method, we also evaluated our intervention approach (including the SSB concept).

Reflection was primarily carried out between the course iterations. Occasional reflective discussions to take note of feedback from mentor meetings in order to plan potential future changes were also conducted while the course was still on-going, but the bulk of the reflection happened after each course. To this end, it is not completely straightforward to draw a clear line between the reflection stage of the preceding cycle and the diagnosis stage of the following cycle in this process.

4.4.2. CAR principles

Below, we describe how the CAR principles and their criteria described by Davison et al. [24] fit into this study. However, Davison et al. [57] recently remark that these "five principles were intended to form the foundation of CAR, with the criteria reflecting specific details that researchers should pay attention to. Recognizing the infinite variety of organizational circumstances, and hence the need for methodological flexibility, adherence to these criteria was never intended to be an absolute or inflexible requirement". In the case of this AR endeavor as well, not all of the criteria are fulfilled, for the most part due to the course setting.

In this study, the project context was the course, and the client organizations were the startups involved in the course. Because of this, it is challenging to fully separate the study of the method from the course from the point of view of some of the principles of CAR [24] and their criteria. Nonetheless, we now reflect the application of each principle and its criteria.

Principle of the Researcher–Client Agreement (RCA). Briefly put, the RCA is about making sure the client understands what is happening and why in the AR process. Particularly from the point of the RCA, it is challenging to separate the project from the method. AR focuses on project contexts and we consider the course to be the project in this case, as it was, indeed, a project-based course. The startup teams were informed, in much detail, about the objectives and the focus of the course itself and how it would proceed, as well as what data would be collected and how it would be analyzed (for grading). In terms of the project, we argue that all the criteria for RCA were fulfilled. To some extent, this also then applies to the method. The StartCards comprise practices that were utilized during the course and the cards were intended, in the context of the course, to help with the completion of these tasks.

However, from the point of view of the method specifically, we did not clearly specify objectives nor discuss data analysis. The startups agreed to having data from the course being used to evaluate the method (except for one team that opted out), but we did not go into specifics about the analysis process. This was done to make sure the teams would not aim to please, given the power dynamic (student/teacher) at play.

Principle of the Cyclical Process Model (CPM). The CPM is about whether the CAR process model was followed as intended and justifying any deviations from it [24]. In this case, we followed the CAR process/CPM, going through the five stages in each cycle. Arguably, some reflection happened during the action taking (the course) as well, when we occasionally discussed data from the mentor meetings between the author team rather than waiting until the end of the AR cycle to conduct such reflection. The bulk of the reflection nonetheless happened at the end of each cycle and before the start of the next cycle, as described in the CPM. This reflection motivated changes to the method or the intervention, which in turn motivated further AR cycles.

While the organizational situation of the startups was analyzed in line with the CPM in the mentor meetings, this had no impact on the actions taken from the point of view of this study. The introduction of the method was pre-planned and proceeded in the planned fashion during the course. The method was the same for all startups during each cycle and the current situation of any one startup during an AR cycle had no impact on the method or how it was introduced. Over the course of the four AR cycles, however, the method was changed in an attempt to make it better suited for the context of these startups as well.

Principle of Theory. The Principle of Theory focuses on the use of theory or existing research in the process [24]. In this case, existing research motivated the method. The content of the cards, as well as the card-based approach itself, are based by existing research (and grey literature). The method was also created to address issues discussed in existing research on software startups. On the other hand, the cards are not based on a single existing theory, and no single theory or model was used to evaluate them.

Principle of Change through Action. The Principle of Change through Action focuses on the interventions: the reasons behind them, their nature and timing, and their impacts [24]. This principle in particular is challenging when it comes to separating the course and the method. The course itself involves actions (course tasks) taken to address the problem (validation). The method is simply introduced to help the client organizations (startups) carry out these tasks. To this end, the mentoring sessions were also intended to help the startups with their project (but discussing them past data related to the method is not in scope of this paper).

The situation of the organizations was assessed through their pitches and the mentor meetings in the beginning of the project, and through their SSBs and the final pitch at the end of the project. Actions were taken in the form of the course tasks and the introduction of the method, which we had designed to address the issue of idea validation. The timing of these actions is clearly documented as a part of the course data. The clients (startups) approved the actions, as the use of the method was voluntary (although the course tasks were mandatory). The motivation of the clients to improve their situation varied. E.g., some startups continued to pursue their original idea even after validation activities had made it seem unfeasible, failing to see the problem.

The Principle of Learning through Reflection. The fifth and final principle of CAR focuses on reflecting on the results and their implications, and then communicating and reporting them [24]. As far as the method is considered, reflection was carried out separately by the researchers and the clients (startups). The startups reflected on the use of the cards in the mentor meetings, the SSBs, and during cycle 4, in the survey. However, this was done primarily as feedback as opposed to active, joint reflection in the form of discussion. I.e., the startup teams provided data that we authors then used to reflect among ourselves. In a typical CAR setting, reflection is, at least to some extent, a more joint activity between the client and the researcher.

In terms of communicating the results, we consider most of the criteria for this principle fulfilled with this paper: the results are reported completely and their implications are considered from the point of view of future research and implications for current software startup research. We have also reflected on the use of CAR itself here in Section 4. Considering the results from the point of view of the project context(s) is carried out through client (startup) self-reporting.

5. Results: Method development through action research

In this section, we report our results. The section is split into four subsections, with each section depicting one AR cycle.

5.1. Cycle 1 (2018)

In 2018, we began to work on StartCards as a result of the *diagnosis* stage of cycle 1. We initially focused on startup failures but zoomed in on validation activities during this process. In this initial diagnosis stage, based on existing literature and our own observations with startups, we considered the lack of validation activities to be a key issue to tackle. Despite lean startup, which at the time had already gained notable traction, emphasizing the importance of validation, in practice startups often failed to carry out sufficient validation. This was also something we had seen happen in the 2017 iteration of the course prior to this study. Based on our diagnosis of the phenomenon on a general level, we began to formulate the cards. The card contents were formulated based on existing software startup research, including some existing studies that some of the authors of the paper have been involved in (e.g., [10,13,50,51]), as well as grey literature (e.g., [4, 14]),

Based on this initial diagnosis of the issue, we devised the initial set of cards as a part of the *action planning* stage of cycle 1. We have discussed how this method was designed in Section 3 (why the card-based approach etc.), and motivated it in Sections 1 and 3. We considered a method to be a suitable way of approaching this issue because we hoped to help software startups work in a more systematic fashion (c.f., [18]). Having devised the method, the focus of the action planning stage of cycle 1 moved to planning the *intervention*, i.e., the introduction of the method during the course. Given the modular, card-based approach of StartCards, we opted to introduce the method in parts.

Table 7
Scratch book data from 2018.

Team	Method use
1	0
2	0
3	1
4	2
5	3
6	3
7	2
8	1
9	0
10	3
11	0

The intervention stage of cycle 1 started in October 2018 when the 2018 course started. The cards were introduced in parts, 1–3 cards per week each week during the 8-week-course. The teams were instructed to utilize the cards as they deemed fit — if at all. The use of the method was voluntary to the teams. The teams were instructed to create some content related to the cards in their scratch books should they utilize them, as one suggestion for scratch book content. Table 7 provides an overview of the SSB data related to the method use (while Table 5 explains the scoring).

The *evaluation* of cycle 1 was carried out using a combination of data from the SSBs and the weekly mentor meetings. The SSBs were studied after the conclusion of the course, as they were produced as the final deliverable. Any relevant observations from the mentor meetings were continuously discussed between the authors. The SSB content of cycle 1 was largely in the form of use notes on the cards, which had been included directly in the SSBs as scans or photographs.

In the *reflection* stage of cycle 1, we made changes to the design of the cards. Already early on in the course, it was concluded that the cards were confusing, as the cards tried to use notation other than natural language, in addition to natural language. Moreover, some of the text on the cards was not considered very practical. As a result, we decided on a three-part layout for the cards that they still use (i.e., motivation, what to do, and common mistakes). This layout split the textual content of the cards into sections, each with its clear purpose. We also attempted to use language that was practitioner-oriented and contained as little academic jargon as possible. The cards seemed to document relevant practices, and seemed to have become more comprehensible during cycle 1: *"We found using the startup cards very easy because they were very much inline with how we were already planning the project and we followed the same process as the cards showed". "We found the cards very similar to using Steve Blanks' customer validation methods alongside Eric Ries' lean startup methodology. The cards could certainly be useful for teams to use as a checklist and a way of reminding what still needs to be done or proceeded further with".*

In addition, based on the activities and progress the teams discussed in the SSBs and the mentoring meetings, two new cards was added. The startup spirit card was created to encourage teams to embrace pivots, as some teams were disheartened by their initial ideas turning out to be unfeasible, occasionally refusing to pivot as a result. In the context of the AR setting course, the card was also intended to encourage teams to work outside the weekly tasks. The Build–Measure–Learn loop was also made into its own card instead of being incorporated into the MVP card for clarity, to further emphasize the idea of building *multiple* MVPs. The teams of 2018 seemed to not have embraced this idea.

Finally, we removed a team-related card. Originally, there were two cards related to the startup team: one for setting up the initial team and one for improving it through team-building. Based on our evaluation of the (lack of) use of these cards, they were combined into one card.

Changes to the method following cycle 1:

- Made card content less confusing by including only natural language
- Decided card layout
- Removed one redundant card by combining two team-related cards into one
- Added 2 new cards for a total of 11
- Made the language on the cards more practitioner-oriented

Table 8
Scratch book data from 2019.

Team	Method use
1	1
2	1
3	3
4	0
5	1
6	0
7	1

5.2. Cycle 2 (2019)

In the *diagnosis* stage of cycle 2, we continued to assess, based on our previous learning in cycle 1 and literature, the problem of validation in early-stage startups. In the *action planning* stages of cycle 2, we added two more cards. To further highlight the importance of using data to validate assumptions, we devised a new card (*Utilize Metrics*) that is based on a grey literature review on startup metrics [51] that we conducted at the end of cycle 1. As one learning from cycle 1, we felt like we should further emphasize the importance of data in order to motivate pivots and validation. We also devised one more new card (*Calculate the Financial Metrics*) with the input of a practitioner expert, an angel investor, with whom we discussed the cards during this time.

Changes to the method during cycle 2 action planning:

- Added 2 new cards for a total of 13

As far as the course was considered, we decided to proceed with the *intervention* in a similar manner as in cycle 1. The cards were introduced in parts, 1–3 cards per week, at the end of each lecture. The cards were handed out physically to the teams, as before. The use of the method continued to be voluntary.

In terms of *evaluation*, cycle 2 focused on mentor meetings, as we ultimately received little data about the cards in the SSBs of 2019. This is illustrated in Table 8. The data simply documented the use of the cards and included little reflection, e.g.: *"Based on the poor answering rate for our questions sent to the companies and people located in [Location], our start-up project noticed the need to pivot. This was also advised in the start-up card, which states that "if surveying potential users indicates that people simply are not interested in your idea, do not hesitate to change it". Our pivoting action was made in relation to the customer segment as well as location".* This prompted reflection about our action taking approach in cycle 2. As before, mentor meetings were conducted on a weekly basis while the SSBs were collected in bulk after the course.

In the *reflection* stage of cycle 2, we considered the data we had collected in cycle 2. While what direct feedback we did receive about the cards in the mentor meetings was positive, the teams were not documenting this use properly. During the reflection stage of cycle 2, we reflected on this in particular. Based on the data we did have, we felt that this might not have been an issue with the method itself, but rather, how it was introduced and how its use was reported. The feedback we had received during the mentor meetings and the SSBs in cycles 1 and 2 pointed to the method being useful and no longer confusing, and yet we received little data about its use in the SSBs in cycle 2. Moreover, some teams seemed to have simply forgotten about the cards they were given during the lectures. Thus, we decided to conduct another AR cycle to gather better data.

5.3. Cycle 3 (2020)

Going into cycle 3, our focus was on improving the intervention based on our reflection in cycle 2. Thus, cycle 3 moved quickly to the *action planning* stage where we considered alternatives as far as the introduction of the method went. We identified three potential problems with our previous interventions: (1) we had not clearly enough suggested the inclusion of feedback related to the cards to be included in the SSBs, (2) we introduced the cards at the end of each lecture, which may have made them feel like an afterthought as they were handed to students already preparing to pack their bags and leave, and (3) we had handed out one card per team, perhaps optimistically expecting them to work primarily face-to-face as a team. However, before the start of the intervention stage of cycle 3, the COVID-19 pandemic forced the course into a fully digital mode.

In terms of *intervention* in cycle 3, even though the course was carried out in a digital format in 2020, it was still organized in the same way and proceeded in the same manner. The only difference was that the lectures and mentor meetings were held online. However, based on our reflection following cycle 2, we made changes to the way we carried out the intervention in cycle 3. The cards were now introduced as a part of the lecture, rather than at the end of each lecture. We also further emphasized that, if the cards were used, reflecting on their use and usefulness would be one good way of producing content for the SSBs. Given that the course was digital, the cards were also digital in cycle 3.

These changes to the intervention seemed to have resulted in far more data on the StartCards in the SSBs. An overview of the SSB data for cycle 3 can be found in Table 9. The *evaluation* in cycle 3 was carried out using this SSB data gathered at the end of the course alongside data from the weekly mentor meetings. We were satisfied with the results of our changes to the intervention in this light. As we reflected on the 2020 data, both the mentor meetings and the scratch books, we were encouraged to not make any major changes to the method based on the positive feedback. Below are some excerpts of positive feedback from different scratch books:

"The start-up cards are useful tools which I will refer to one day if I have an idea for a business".

"We used the help of the startup cards to come up with ideas on how we can map the market and see whether the idea is actually valid and there is a market need. With the help of the startup cards, we got the idea to search the internet to find useful information about the market and competition, and we also got the idea to create a small questionnaire to see if there really is a market pain".

"Also business cards that we are using in the course seems to be a very great tool for startups, and I would save the cards for future usage"

In terms of negative feedback, the cards seemed to be least utilized by teams (or individuals) who felt that they were already familiar with the practices on the cards. We considered this a positive issue, as we considered it to also mean that the cards succeeded in conveying good practices that students with previous startup experience were occasionally familiar with. The most widely used card was the card discussing pitching as a practice, which had also been the most commonly used card in cycle 2. Based on the feedback from cycle 3, we did not find any glaring issues with the existing cards in terms of wording, formatting, or how the method was presented, and decided to focus on polishing the content of the cards and making sure the card deck covered all the topics we considered relevant based on literature.

However, as we reflected on the content of the SSBs in relation to the content of the method, we decided to add one more card. The method had previously not featured a card related to business models. Devising one using the Business Model Canvas (BMC) had been a course task, though, and as such it had been overlooked from the point of view of the cards themselves. We felt this needed to be addressed to make the cards independent of the course, and added such a card. We decided to start one last AR cycle to validate this card and collect more data, and to perhaps consider the addition of other cards.

Changes to the method based on cycle 3:

Table 9
Scratch book data from 2020.

Team	Method Use
1	2
2	3
3	3
4	2
5	3
6	3
7	1
8	3
9	3
10	2
11	1
12	3

- Added 1 card for a total of 14
- Iteratively continued to improve wording on the cards

5.4. Cycle 4 (2021)

In the *diagnosis* stage of cycle 4, we continued to study literature on startup failures in order to understand what could perhaps still be added to the method in addition to the card on business models we had added. We considered the addition of some cards to further tackle some of the anti-patterns of Klotins et al. [8]. As a part of the *action planning* of cycle 4, we drafted some new cards based on these anti-patterns. However, they were more relevant for software startups further along in their product development process or life cycle 1, and were consequently not within the scope of the course. As a result, we had problems incorporating them into our action plans for cycle 4, and ultimately they were not introduced in the course.

After cycle 3, we saw no need to change the way the method was introduced to the course anymore after making changes to it in cycle 3. For the most part, the *intervention* in cycle 4 also proceeded in the same fashion as in cycle 3. However, the course started with a new type of hackathon-like event, which was prepared for educational reasons rather than as a part of this AR endeavor. In this hackathon, the students formed teams and prepared a business idea to address the need of a real customer in one day and pitched the idea at the end of the afternoon. Each team was handed the full StartCards deck for this event. Following this event, the course proceeded in the usual fashion in terms of card introduction, with 1–3 cards being introduced every week.

In terms of *evaluation*, we continued to utilize data from the SSBs, collected after the end of the course, and the weekly mentor meetings. An overview of the SSB data from 2021 can be found in Table 10. However, in addition to these two types of data, we also utilized survey data from the post-course survey (Table 4) in this cycle.

The SSBs, where they involved feedback about the cards, contained predominantly positive remarks. Especially the earlier cards were considered useful by the teams who felt lost early on in the process of starting a startup. They felt that the cards provided them with a list of things to do, which helped them clarify the process. Below is an example of the reflection in one of the 2021 SSBs:

"For me I will start my preparation by having a look at the start up cards. Appealing the idea card, we already discussed it during the team meeting. The great pitch card, I like it so much it gave me insightful ideas about our pitch next Monday. We may ask a short question about sleeping problems and if anyone is using therapy lights to get yes or no answers because of the short pitch time. I like the idea of depending on the audience to complete our team. Generally, all the cards are informative and important".

As for the survey, one respondent mentioned not utilizing the cards, but provided no reasoning behind this. However, based on the SSB data (Table 10), other members of the team must have nonetheless used them. Another respondent considered the method to not be very

Table 10
Scratch book data from 2021.

Team	Method use
1	1
2	2
3	1
4	1
5	1
6	1
7	2
8	2
9	2
10	3
11	1
12	3
13	1

useful, remarking that *"Most work was done without cards. But without them things would be worse"*. All other teams seemed to consider them useful to a varying degree based on the survey responses.

In their responses to the survey, the respondents discussed the role of the cards as checklists and pointers for things they needed to work on and regarded their contents as helpful in pointing them to the right direction. No feedback necessitating changes to the method was received in the survey, although one respondent reflected on the contents of the cards and felt that they had felt more relevant from the point of view of future progress rather than the current state of their startup. Below are some direct examples of the feedback gathered through this brief survey:

"Yes, they give good tips and some of them worked excellently as reminders on some topics that I already knew but needed to be refreshed on. Also, as some of the members may not have known about some topics like what a pitch includes, the cards explain them well so there was less work for me in introducing the ideas when needed".

"Yes, they were useful, since they helped explaining the important aspects of each stage of the project in a concise way".

"Yeah, they helped us think about aspects that are not maybe first in mind when coming up with business idea. They helped us deepen the idea and sort of make it more stable"

One of the teams remarked that the course setting itself made the method less useful. Specifically, they felt that the lectures and weekly assignments occasionally discussed the card contents as well, making it so that the method could end up feeling supplementary:

"I think they were somewhat useful as reminders of core things that we should focus on, I liked their compact form. Probably we would have found them more useful if there hadn't been weekly lectures where lecturers explicitly told us what to do. If I were doing a real startup I would probably make giant posters out of them and put them on office walls".

This survey data, in conjunction with the data from the 2021 SSBs, points towards the method being useful for early-stage startups in its current state. No further changes to the method were made based on the data from cycle 4. Consequently, this marked the exit from the cyclical AR process.

6. Discussion

StartCards is a method for early-stage software startups. Existing methods are poorly suited for software startups, as startups operate in a context different from traditional software organizations [15,17,20,58]. Startup methods are scarce, with the lean startup [14] being the most prominent one. However, the lean startup is difficult to utilize in practice [20], resulting in various tools, guides, and books to help implement it (e.g., Running Lean by Maurya [21]).

The method proposed in this paper provides one potential option for tackling this issue. It presents a novel contribution by proposing a startup method based on both grey literature and scientific literature, including practices discussed in the lean startup. This attempt at bridging the gap between research and practice in the area of software startup research is the primary theoretical contribution of this paper. In this regard, the paper helps address the problem of insufficient validation in software startups discussed in extant literature [8]. By using the method, early-stage startups may be made aware of the importance of validation activities and are familiarized with good practices for validating their business and product idea.

The cards that comprise the method are based on both existing academic research and practitioner literature. The practices present in the cards are practices widely discussed in literature, and they include the well-known lean startup [14] practices as well (MVP, pivot, and BML loop). Aside from simply utilizing existing literature, we have also carried out relevant studies ourselves in the past (including, e.g., [10,13,50,51]), which have supported the design and development of the method. In this regard, the contribution of the method is the way it attempts to highlight the acknowledged importance of these practices for practitioners, encouraging their use already early on in the startup process in order to avoid situations where products are developed without market potential. The contribution of the paper also begins to answer the call of Unterkalmsteiner et al. [15] for methods aimed at software startups. At the same time, it also provides a contribution in terms of operationalizing lean startup principles, which existing research argues can be difficult to utilize in practice [20].

After cycle 3, we considered adding more cards to the method based on the anti-patterns of Klotins et al. [8]. We created drafts of three additional cards intended to tackle some of these anti-patterns and considered including them into method following cycle 3. However, these have not been not utilized in the AR process of this paper due to being a poor fit for the course, and also based on our consideration of the review feedback, have not been included into the method. We feel that this makes the scope of the method clearer, as these planned cards were more suited for more advanced startups in the context of the life cycle model discussed by Wang et al. [7] (Fig. 1). Thus, to not overscope the method and to keep it aimed at early-stage startups specifically, we consider the method suitable for its intended purpose with the existing 14 cards.

6.1. Practical implications

The primary practical implication of this paper is the method itself. The data we have collected so far suggests that early-stage startups consider the method useful in its current form. Specifically, it is useful of keeping track of things to focus on. The contents of the cards act as checklists for things to do, and the cards themselves highlight practices that should be carried out in early-stage startups.

To make the cards useful, the following design philosophy was used while designing them. First, they should be lightweight. If organizations use methods, they prefer lightweight ones [59]. Startups in particular seldom use heavy processes [17], although this is likely to also be because they consider the existing ones poorly suited for their context (e.g., agile ones [20]). Secondly, the cards are method-agnostic. As startups are known to prefer working using individual agile practices or even ad hoc [17,27,28], the method has been designed to not impose a heavy process that would be unsuited for use with a wide variety of work processes. Instead, the cards encourage adding new practices into existing work processes. Thirdly, as iterative development is the norm in SE, as well as lean startup [14], the cards support iterative work. The cards are largely designed to be stand-alone, with each one alone forming a coherent whole. As such, a team can select which cards they feel are most relevant to their current situation and re-evaluate this as the situation changes.

As the method was evaluated in a practical course setting, this paper also has practical implications for startup education. In this paper, the cards form a process that was used to guide the creation of early-stage startups starting from initial ideation and team-forming in a practical course setting. Aside from serving as a method for early-stage

startups, the cards can also serve as a framework for teaching startup entrepreneurship. The topics covered in the cards do not form a strictly linear process, but the cards, as they are ordered, can be used as a guideline for topics to discuss in such a course. They have been ordered with startup life cycle frameworks (e.g., [7,44]) in mind, and as such earlier cards are more relevant in the earlier stages of the startup life cycle.

6.2. Threats to validity

In this section, we discuss the validity threats of this paper. As a framework, we utilize the four types of validity threats in SE discussed by Runeson & Höst [60]: construct validity, internal validity, external validity, and reliability.

6.2.1. Construct validity

In relation to constructs, we would like to discuss practices as a concept in SE. The cards can be considered to depict practices. While the concept of a practice is clear (i.e., a practice describes how work can, or should be, be carried out), the level of detail or abstraction when describing practices is arguably subjective. Many practices could be further split into more atomic practices for clarity. The ones included in StartCards are rather general practices. In fact, we could describe many of them as 'patterns' rather than practices. For example, idea validation can be seen as a practice, but there are various more specific practices (e.g., user interview, MVP, surveys etc.) that can be used to carry out validation activities, and not all of these were included as cards. Idea validation practices could include the use of HyMap [36], which is a tool designed to help startups validate assumptions by turning them into hypotheses, and in the paper the authors discuss in detail how it can be utilized. In the cards, we aim to highlight that the cards are hardly all-encompassing, and that their users should look into more specific practices as necessary. To this end, HyMap is mentioned as one option in the cards as well.

6.2.2. Internal validity

The most prominent threat to internal validity in this case is that we are unable to show the effects of the method based on our current data. E.g., we cannot claim that the method increases the likelihood of success in early-stage startups, or otherwise has a positive effect on them by some metric. Aside from 'success' being difficult to determine and measure, this is also a shortcoming of our research approach. For the time being, we have not, e.g., conducted A–B testing that would let us see how the use of the method might (or might not) positively impact early-stage startups compared to teams that do not use it. The current data we have utilized has been based on self-reporting.

Ensuring business success is a lofty and practically most likely an impossible goal to achieve given the sheer number of factors at play. On the other hand, the method does place emphasis on validation (or, in the context of traditional SE, RE) activities. The focus is thus on encouraging startup teams to test their assumptions about their business, and to fail fast if failure is imminent, as opposed to spending time and effort developing a service no one will ever use (as e.g., Klotins et al. [8] argue is a common issue in software startups). Some of the feedback we have discussed in Section 5 indicates that the method is useful for this purpose, but, again, it is reliant on self-reporting.

To some extent, we argue that the way we have designed the method could help provide some validity in this regard. We have based the practices described on the cards of the method on existing research, as well as grey literature. Each practice, such as pivoting (e.g., [12,13]), has an academic research background (as well as one in grey literature). The purpose of the cards is to emphasize the potential importance of these practices, and to encourage the teams to look further into these practices. However, how they are then utilized in practice by the startup can vary greatly between startups.

6.2.3. External validity

AR typically utilizes many different types of data, which results in a large number of external validity threats. While we have had, we argue, a sufficient number of startup teams ($n = 44$) involved in the process, there are various other threats to external validity at play here.

Our most prominent source of data have been the SSBs. This is a novel data collection approach, which on its own presents threats to validity. Although for the purposes of this paper, these can be treated as learning diaries or experience reports, as we only included the contents related to the use of the method. In hindsight, we can highlight three potential shortcomings with our data collection approach: (1) the use of the method being voluntary in all AR cycles, (2) us failing to urge the teams strongly enough to include data about the use of the method into their learning diaries (SSBs) in cycle 1 and 2, and (3) the SSBs being produced by teams rather than individuals.

We chose to make the use of the method voluntary in an attempt to avoid bias, so that the student teams would not aim to please with their responses. We were concerned that the students might be unlikely to be honest about disliking the method before receiving their grades, for example. Yet, as we relied on self-reporting, we have little data from teams that did not utilize the method. Though the data we did collect, we argue, is more likely to include honest feedback, we wanted more data. Similarly, not all teams included notes about the use of the method in their SSB despite using it, which led to us changing our approach slightly after cycle 2.

Thus, after receiving particularly little learning diary (SSB) data in cycle 2, we began to place more emphasis on the method during the lectures, though while nonetheless keeping its use voluntary. We also urged teams to include more personal reflection in their SSBs in general after cycle 2, as we felt that there was too little of it overall, and not just in relation to the use of StartCards. This lack of SSB data in cycle 2 contributed to our decision to continue the AR process and resulted in more data for cycles 3 and 4 (Tables 9 and 10).

Additionally, the SSBs being produced by teams rather than individuals presents a validity threat. The contents of the SSB may not always reflect the collective thoughts of the team, but rather just the one(s) writing it. For example, a team may opt to have one person focus on producing and editing the SSB, while the rest of the team focuses on other tasks. We attempted to tackle this limitation to some extent by asking each team member to include some personal reflection as well, especially after cycle 2. This is also a limitation for the survey conducted at the end of cycle 4, as the responses were from individuals as opposed to collective responses from teams.

To alleviate these limitations related to the SSB data, we utilized other types of data as well. Data was also collected from weekly mentor meetings with the teams, which provided us with an additional source of supplementary data on the use of the method. However, these discussions were not recorded and transcribed. In 2021, additional data from a brief survey was used to further support the SSB data as well.

We also highlight the use of student data as a limitation. However, in this case, we argue that our approach and research area also lessen the threats to validity stemming from the use of student data. Whereas students are less analogous to, e.g., senior developers, startup practitioners are not necessarily as far from students demographically (e.g., many startup teams are inexperienced [17,27]). It is also not uncommon for startups to be founded by students [61]. In fact, many software startups are founded by students. Moreover, the course in question was a learning "through" entrepreneurship [26] course where the student teams worked on startups as though they were real, while some were intended to be real startups. The only difference between the real startups and the simulated ones were the motivations of the teams. Even the teams who never intended for the startup to become a real business still interacted with real customers, developed a real MVP, and in general carried out 'real' startup activities.

Furthermore, we would also point out that we wanted to study early-stage startups in particular, as the method is aimed at such. Finding very early-stage startups that are still in the process of coming up

with their initial idea can be difficult. For example, most startups found in startup ecosystems, such as the ones found in startup incubators, are typically already further along in the startup life cycle. Thus, despite its limitations, this approach, to its merit, has made it possible for us to gather data from early-stage startups.

6.2.4. Reliability

Reproducing AR studies can be difficult as they target organizations and change the organization involved during the study, making the use of the same case organizations impossible. However, as we have had a moderate number of startups involved in the study ($n = 44$), similar studies in a similar setting should see similar results in terms of feedback on the method. Rather than reproducing the study here, though, it might be more beneficial for further studies to take the method outside the course setting, as is our plan in the future as well.

7. Conclusions

In this paper, we have presented a card-based method for software startups: StartCards. The method is intended to help early-stage software startups validate their ideas, as insufficient validation continues to be an issue in software startups [8,36]. The method consists of a set of 14 cards. Each of the cards describes startup practices based on academic and practitioner literature. The cards are intended to help early-stage software startups grasp important startup practices, and especially on encouraging startups to validate their business ideas and assumptions in order to avoid situations where a software with no market interest is developed.

The method was developed iteratively using an action research approach. In this paper, we have detailed the four action research cycles used to develop the method through Canonical Action Research [24]. In the process, the method was developed in a practical startup entrepreneurship course setting over four years, while involving 44 startup teams in total. Data was collected through modified learning diaries called Startup Scratch Books (SSBs), weekly mentor meetings with the startup teams, and, during the final cycle, through a post-course survey related to the use of the method.

The method presents a novel contribution in startup literature, as existing startup methods in academic literature are scarce. It answers the call of Unterkalmsteiner et al. [15] for methods aimed at software startups. The method aims to bring some of the insights from academic literature into practice, while also drawing from grey literature. This paper presents the first published version of the method. The method can be found on figshare: https://doi.org/10.6084/m9.figshare.20151722.v1.

7.1. Future research

Further studies are needed to determine whether the method has an impact on success, or some other type of positive impact on some other metric in software startups. So far, the method has been developed using data from use experiences of the method, and specifically by relying on self-reporting from the startup teams involved. While the teams using the method seem to consider it useful, different types of data would be needed to further study its impacts on their work, and possibly success. We also recommend keeping this in mind if utilizing the method.

The existing set of 14 cards could also be extended or modified in some way, e.g., based on such evaluation. However, while additional cards could be added to the method, we feel that it would perhaps be more appropriate to develop other methods if the planned extension is particularly large. This particular method is aimed at early-stage software startups who are in the concept of early development stages (in the context of the life cycle model discussed in [7] and in Fig. 1 of this paper). Startups face various challenges past these initial stages as well [7,8]. Given the present lack of methods tailored for startups, further tooling and methods are arguably still needed to address these anti-patterns and challenges later in the startup life cycle as well. Especially methods focused on the unique aspects of software development in startups, e.g., as discussed and conceptualized by Giardino et al. [18] in their Greenfield Startup Model, are needed. However, to achieve this, further research into understanding software development may also still be needed to better understand what kind of software development processes startups would favor.

Finally, StartCards was developed in a practical course setting. It is also an educational tool that can be used in learning "through" entrepreneurship type courses [26] in startup education. Future studies might be interested in looking further into the role of StartCards as an educational tool. The modified learning diaries (SSBs) that we have used to collect much of our data in this paper may also be interesting tools for startup education, as we have originally developed the concept for tackling our own pedagogical needs and interests (see [54]).

CRediT authorship contribution statement

Kai-Kristian Kemell: Writing – original draft, Writing – review & editing, Formal analysis, Methodology, Conceptualization. **Anh Nguyen-Duc:** Supervision, Methodology, Conceptualization. **Mari Suoranta:** Supervision, Project administration, Data curation, Conceptualization, Resources. **Pekka Abrahamsson:** Supervision, Project administration, Data curation, Methodology, Conceptualization, Resources, Formal analysis.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.infsof.2023.107224.

Data availability

The authors do not have permission to share data

Acknowledgments

We would like to express our gratitude to the students and the teaching assistants of the Venture Lab & Lean Startup course for their efforts during the multiple instances of the course over these four years, and thus also their contribution to the development of the StartCards. We thank the reviewers for helping us notably improve this paper with their comments, feedback, and suggestions during the review process.

References

[1] The State of European Tech 2020, https://2020.stateofeuropeantech.com.
[2] Pitchbook, 4Q 2018 PitchBook-NVCA venture monitor, 2019, https://pitchbookcom/news/reports/4Q2018-pitchbook-nvca-venture-monitor.
[3] M. Crowne, Why software product startups fail and what to do about it. Evolution of software product development in startup companies, in: IEEE International Engineering Management Conference, Vol. 1, 2002, pp. 338–343, http://dx.doi.org/10.1109/IEMC.2002.1038454.
[4] S. Blank, Four Steps to Epiphany, K&S Ranch, 2013.
[5] C. Giardino, X. Wang, P. Abrahamsson, Why early-stage software startups fail: A behavioral framework, in: International Conference of Software Business, ICSOB, 2014, pp. 27–41.
[6] J. Mullins, R. Komisar, Getting to Plan B: Breaking Through to a Better Business Model, Harvard Business Review Press, 2009.
[7] X. Wang, H. Edison, S.S. Bajwa, C. Giardino, P. Abrahamsson, Key challenges in software startups across life cycle stages, in: H. Sharp, T. Hall (Eds.), Agile Processes, in Software Engineering, and Extreme Programming, Springer International Publishing, Cham, 2016, pp. 169–182.
[8] E. Klotins, M. Unterkalmsteiner, T. Gorschek, Software engineering antipatterns in start-ups, IEEE Softw. 36 (2) (2019) 118–126, http://dx.doi.org/10.1109/MS.2018.227105530.
[9] F. Kon, D. Cukier, C. Melo, O. Hazzan, H. Yuklea, A panorama of the Israeli software startup ecosystem, SSRN (2014) http://dx.doi.org/10.2139/ssrn.2441157.

[10] A.N. Duc, P. Abrahamsson, Minimum viable product or multiple facet product? The role of MVP in software startups, in: H. Sharp, T. Hall (Eds.), Agile Processes, in Software Engineering, and Extreme Programming, Springer International Publishing, Cham, 2016, pp. 118–130.

[11] V. Lenarduzzi, D. Taibi, MVP explained: A systematic mapping study on the definitions of minimal viable product, in: 2016 42th Euromicro Conference on Software Engineering and Advanced Applications, SEAA, 2016, pp. 112–119, http://dx.doi.org/10.1109/SEAA.2016.56.

[12] S. Bajwa, X. Wang, A. Nguyen Duc, P. Abrahamsson, How do software startups pivot? Empirical results from a multiple case study, in: International Conference of Software Business, 2016, pp. 169–176.

[13] S.S. Bajwa, X. Wang, A. Nguyen-Duc, P. Abrahamsson, "Failures" to be celebrated: an analysis of major pivots of software startups, Empir. Softw. Eng. 22 (2017).

[14] E. Ries, The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses, Crown Business, New York, 2011.

[15] M. Unterkalmsteiner, P. Abrahamsson, X. Wang, A. Nguyen-Duc, S. Shah, S. Bajwa, G. Baltes, K. Conboy, E. Cullina, D. Dennehy, H. Edison, C. Fernandez-Sanchez, J. Garbajosa, T. Gorschek, E. Klotins, L. Hokkanen, F. Kon, I. Lunesu, M. Marchesi, L. Morgan, M. Oivo, C. Selig, P. Seppänen, R. Sweetman, P. Tyrväinen, C. Ungerer, A. Yagüe, Software startups - a research agenda, e-Informatica Softw. Eng. J. 10 (1) (2016) 89–123, http://dx.doi.org/10.5277/e-Inf160105, EXT="Shah, Syed".

[16] G. Wang, J. Nandhakumar, Strategic swaying: How startups grow digital platforms, in: ICIS 2017 Proceedings, 2017.

[17] N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, P. Abrahamsson, Software development in startup companies: A systematic mapping study, Inf. Softw. Technol. 56 (10) (2014) 1200–1218, http://dx.doi.org/10.1016/j.infsof.2014.04.014, URL https://www.sciencedirect.com/science/article/pii/S0950584914000950.

[18] C. Giardino, N. Paternoster, M. Unterkalmsteiner, T. Gorschek, P. Abrahamsson, Software development in startup companies: The greenfield startup model, IEEE Trans. Softw. Eng. 42 (6) (2016) 585–604.

[19] A. Nguyen-Duc, K.-K. Kemell, P. Abrahamsson, The entrepreneurial logic of startup software development: A study of 40 software startups, Empir. Softw. Eng. 26 (2021) http://dx.doi.org/10.1007/s10664-021-09987-z.

[20] J. Bosch, H. Holmström Olsson, J. Björk, J. Ljungblad, The early stage software startup development model: A framework for operationalizing lean principles in software startups, in: B. Fitzgerald, K. Conboy, K. Power, R. Valerdi, L. Morgan, K.-J. Stol (Eds.), Lean Enterprise Software and Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 1–15.

[21] A. Maurya, Running Lean: Iterate from Plan a to a Plan that Works, O'Reilly Media Inc., 2012.

[22] E. Klotins, M. Unterkalmsteiner, T. Gorschek, Software engineering in start-up companies: An analysis of 88 experience reports, Empir. Softw. Eng. 24 (1) (2019) 68–102.

[23] S. Bajwa, X. Wang, A. Nguyen Duc, R. Chanin, R. Prikladnicki, L. Pompermaier, P. Abrahamsson, Start-ups must be ready to pivot, IEEE Softw. 34 (2017) 18–22, http://dx.doi.org/10.1109/MS.2017.84.

[24] R. Davison, M.G. Martinsons, N. Kock, Principles of canonical action research, Inf. Syst. J. 14 (1) (2004) 65–86, http://dx.doi.org/10.1111/j.1365-2575.2004.00162.x.

[25] G.I. Susman, R.D. Evered, An assessment of the scientific merits of action research, Adm. Sci. Q. 23 (4) (1978) 582–603, URL http://www.jstor.org/stable/2392581.

[26] F. Sirelkhatim, Y. Gangi, Entrepreneurship education: A systematic literature review of curricula contents and teaching methods, Cogent Bus. Manag. 2 (2015) 1052034, http://dx.doi.org/10.1080/23311975.2015.1052034.

[27] C. Giardino, M. Unterkalmsteiner, N. Paternoster, T. Gorschek, P. Abrahamsson, What do we know about software development in startups? IEEE Softw. 31 (5) (2014) 28–32.

[28] J. Pantiuchina, M. Mondini, D. Khanna, X. Wang, P. Abrahamsson, Are software startups applying agile practices? The state of the practice from a large survey, in: 18th International Conference on Agile Software Development (XP 2017), 2017, pp. 167–183.

[29] Digital.ai, 14th Annual State of Agile Report, 2020, https://digital.ai/catalyst-blog/the-14th-annual-state-of-agile-report.

[30] V. Berg, J. Birkeland, A. Nguyen-Duc, I. Pappas, L. Jaccheri, Software startup engineering: a systematic mapping study, J. Syst. Softw. 144 (2018) 255–274.

[31] P. Seppänen, K. Liukkunen, M. Oivo, Little big team: Acquiring human capital in software startups, in: Proceedings of the 18th International Conference on Product-Focused Software Process Improvement, PROFES 2017, Innsbruck, Austria, November 29–December 1, 2017, pp. 280–296.

[32] P. Seppänen, Yes, we can! building a capable initial team for a software startup, in: Fundamentals of Software Startups: Essential Engineering and Business Aspects, Springer International Publishing, Cham, 2020, pp. 45–59.

[33] K. Kemell, A. Elonen, M. Suoranta, A. Nguyen-Duc, J. Garbajosa, R. Chanin, J. Melegati, U. Rafiq, A. Aldaeej, N. Assyne, A. Sales, S. Hyrynsalmi, J. Risku, H. Edison, P. Abrahamsson, Business model canvas should pay more attention to the software startup team, in: 2020 46th Euromicro Conference on Software Engineering and Advanced Applications, SEAA, IEEE Computer Society, Los Alamitos, CA, USA, 2020, pp. 342–345, http://dx.doi.org/10.1109/SEAA51224.2020.00063.

[34] C. Seaman, Y. Guo, Measuring and monitoring technical debt, Adv. Comput. 82 (2011) 25–46.

[35] M. Kuhrmann, P. Tell, R. Hebig, J.A.-C. Klunder, J. Munch, O. Linssen, D. Pfahl, M. Felderer, C. Prause, S. Macdonell, J. Nakatumba-Nabende, D. Raffo, S. Beecham, E. Tuzun, G. Lopez, N. Paez, D. Fontdevila, S. Licorish, S. Kupper, G. Ruhe, E. Knauss, O. Ozcan-Top, P. Clarke, F.H. Mc Caffery, M. Genero, A. Vizcaino, M. Piattini, M. Kalinowski, T. Conte, R. Prikladnicki, S. Krusche, A. Coskuncay, E. Scott, F. Calefato, S. Pimonova, R.-H. Pfeiffer, U. Pagh Schultz, R. Heldal, M. Fazal-Baqaie, C. Anslow, M. Nayebi, K. Schneider, S. Sauer, D. Winkler, S. Biffl, C. Bastarrica, I. Richardson, What makes agile software development agile, IEEE Trans. Softw. Eng. (2021) 1, http://dx.doi.org/10.1109/TSE.2021.3099532.

[36] J. Melegati, E. Guerra, X. Wang, Hymap: Eliciting hypotheses in early-stage software startups using cognitive mapping, Inf. Softw. Technol. 144 (2022) 106807, http://dx.doi.org/10.1016/j.infsof.2021.106807, URL https://www.sciencedirect.com/science/article/pii/S095058492100241X.

[37] P. Zave, Classification of research efforts in requirements engineering, ACM Comput. Surv. 29 (4) (1997).

[38] B. Nuseibeh, S. Easterbrook, Requirements engineering: A roadmap, in: ICSE '00: Proceedings of the Conference on the Future of Software Engineering, 2000, pp. 35–46.

[39] I. Inayat, S.S. Salim, S. Marczak, M. Daneva, S. Shamshirband, A systematic literature review on agile requirements engineering practices and challenges, Comput. Hum. Behav. 51 (2015) 915–929.

[40] J. Melegati, A. Goldman, F. Kon, X. Wang, A model of requirements engineering in software startups, Inf. Softw. Technol. 109 (2019) 92–107.

[41] G. Kotonya, I. Sommerville, Requirements Engineering: Processes and Techniques, John Wiley & Sons, Inc., New York, NY, USA, 1998.

[42] B.H.C. Cheng, J.M. Atlee, M. Joanne, Research directions in requirements engineering, in: Proceedings of FOSE '07 2007 Future of Software Engineering, 2007, pp. 285–303.

[43] A. Osterwalder, Y. Pigneur, Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers, John Wiley & Sons, 2010.

[44] A. Nguyen-Duc, S.M.A. Shah, P. Ambrahamsson, Towards an early stage software startups evolution model, in: 2016 42th Euromicro Conference on Software Engineering and Advanced Applications, SEAA, 2016, pp. 120–127, http://dx.doi.org/10.1109/SEAA.2016.21.

[45] I. Jacobson, R. Stimson, Tear down the method prisons! set free the practices! essence: a new way of thinking that promises to liberate the practices and enable true learning organizations, Queue 16 (5) (2018) 101–127.

[46] S.A. Alvarez, J.B. Barney, Discovery and creation: alternative theories of entrepreneurial action, Strategic Entrepreneurship J. 1 (1–2) (2007) 11–26, http://dx.doi.org/10.1002/sej.4.

[47] J. Moss, How to make a winning pitch deck for your startup, 2018, https://www.forbes.com/sites/forbesnycouncil/2018/08/28/how-to-make-a-winning-pitch-deck-for-your-startup/.

[48] C. Clark, The impact of entrepreneurs' oral 'pitch' presentation skills on business angels' initial screening investment decisions, Venture Capital 10 (3) (2008) 257–279, http://dx.doi.org/10.1080/13691060802151945.

[49] I. Jacobson, P.-W. Ng, P. McMahon, I. Spence, S. Lidman, The essence of software engineering: The SEMAT kernel: A thinking framework in the form of an actionable kernel, Queue 10 (10) (2012) 40–51, http://dx.doi.org/10.1145/2381996.2389616.

[50] K.-K. Kemell, V. Ravaska, A. Nguyen-Duc, P. Abrahamsson, Software startup practices – software development in startups through the lens of the essence theory of software engineering, in: M. Morisio, M. Torchiano, A. Jedlitschka (Eds.), Product-Focused Software Process Improvement, Springer International Publishing, Cham, 2020, pp. 402–418.

[51] K.-K. Kemell, W. Xiaofeng, A. Nguyen-Duc, J. Grendus, T. Tuunanen, P. Abrahamsson, 100+ Metrics for software startups : A multi-vocal literature review, in: SiBW 2018: Proceedings of the First International Workshop on Software-Intensive Business: Start-Ups, Ecosystems and Platforms, 2018, pp. 15–29.

[52] C. Zott, R. Amit, L. Massa, The business model: Recent developments and future research, J. Manag. 37 (4) (2011) 1019–1042, http://dx.doi.org/10.1177/0149206311406265.

[53] E. Rasmussen, R. Sørheim, Action-based entrepreneurship education, Technovation 26 (2006) 185–194.

[54] P. Abrahamsson, M. Suoranta, S. Lahti, K.-K. Kemell, The startup scratch book – opening the black box of startup education, in: E. Klotins, K. Wnuk (Eds.), Software Business, Springer International Publishing, Cham, 2021, pp. 193–200.

[55] D. Remenyi, Innovation in Teaching of Research Methodology Excellence Awards 2020, Acpil, 2020.

[56] J. Moon, Learning Journals: A Handbook for Reflective Practice and Professional Development, Routledge, London, 2006.

[57] R.M. Davison, M.G. Martinsons, J. Malaurent, Research perspectives: Improving action research by integrating methods, J. Assoc. Inf. Syst. 22 (2021).

[58] C. Giardino, S. Bajwa, X. Wang, P. Abrahamsson, Key challenges in early-stage software startups, in: Lecture Notes in Business Information Processing, Vol. 212, 2015, pp. 52–63.

[59] P. Abrahamsson, N. Iivari, Commitment in software process improvement - in search of the process, in: Proceedings of the 35th Annual Hawaii International Conference on System Sciences, 2002, pp. 3239–3248, http://dx.doi.org/10.1109/HICSS.2002.994403.

[60] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empir. Softw. Eng. 14 (2008) http://dx.doi.org/10.1007/s10664-008-9102-8.

[61] S. Xu, How students are founding, funding, and joining startups, Techcrunch (2019) URL https://techcrunch.com/2019/02/06/how-students-are-founding-funding-and-joining-startups.