

Are Software Startups Applying Agile Practices? The State of the Practice from a Large Survey

Jevgenija Pantiuchina¹, Marco Mondini¹, Dron Khanna¹, Xiaofeng Wang^{1(✉)},
and Pekka Abrahamsson²

¹ Free University of Bozen-Bolzano, Piazza Domenicani 3, 39100 Bolzano, Italy
{Jevgenija.Pantiuchina, marco.mondini, dron.khanna, xiaofeng.wang}@unibz.it

² Norwegian University of Science Technology, 7491 Trondheim, Norway
pekkaa@ntnu.no

Abstract. Software startups operate under various uncertainties and the demand on their ability to deal with change is high. Agile methods are considered a suitable and viable development approach for them. However, the competing needs for speed and quality may render certain agile practices less suitable than others in the startup context. The adoption of agile practices can be further complicated in software startups that adopt the Lean Startup approach. To make the best of agile practices, it is necessary to first understand whether and how they are used in software startups. This study targets at a better understanding of the use of agile practices in software startups, with a particular focus on lean startups. Based on a large survey of 1526 software startups, we examined the use of five agile practices, including quality related (regular refactoring and test first), speed related (frequent release and agile planning) and communication practice (daily standup meeting). The findings show that speed related agile practices are used to a greater extent in comparison to quality practices. Daily standup meeting is least used. Software startups who adopt the Lean Startup approach do not sacrifice quality for speed more than other startups do.

Keywords: Software startups · Agile practice · Lean startup · Minimum viable product · Pivot · Quality vs Speed

1 Introduction

Startups are organizations designed to create new products or services under the conditions of extreme uncertainty, which constantly seek repeatable, profitable and scalable business models and aim at rapid growth [1, 2]. Software startups are startups that have a primary focus on developing new and innovative software-intensive products or services from which business value is created. Even though sharing common characteristics with other types of startups, such as resource scarcity and a lack of operational history [3], software startups are often caught up in the wave of technological change frequently happening in software industry, such as new computing and network technologies and devices [4].

As the ability to accommodate frequent change is essential in the startup context, agile methods have been considered the most suitable process model since they enable software startups to embrace change, and allow development to adapt to business strategies [5]. Fast release with an iterative and incremental approach shortens the lead time from idea conception to production to market, which is especially important for software startups as “done is better than perfect” and “move fast and break things” are the slogans or mantras that they follow in order to respond to the challenges they are confronted with [6].

However, since software startups are constantly under the huge pressure of time-to-market and need to move really fast, product quality may be treated with a low priority and technical debt is accumulated to gain the speed to market [7]. As a result, certain agile practices that ensure the quality of software, such as refactoring and test-driven development, may not be considered viable practices for software startups, especially at the early stages [8]. But the accumulated technical debt, if not paid back in time, will eventually slow down the development speed [7], which means software startups cannot afford to ignore quality and related engineering practices as they progress through the stages of development.

The adoption of agile practices can be further complicated when software startups follow the Lean Startup approach to develop their business, which puts even more emphasis on quick prototyping [9], testing prototypes with potential customers, and getting early feedback. The use of Lean Startup approach may intensify the so-called “developers dilemma”—the balancing act between quality and speed to achieve fast product iteration [10], and render the agile practices related to quality even less viable to software startups.

To understand how software startups can better use and benefit from different agile practices for their needs for quality and speed, it is important to understand firstly if and how software startups are currently using agile practices. The existing software engineering literature has accumulated a growing body of knowledge on the application of agile methods in established companies, large or small. However it casts very few lights on the use of agile practices in software startups, let alone in the startups who adopt the Lean Startup approach. Based on this observation, the study presented in this paper targets at understanding the state of the practice of agile practices in software startups, and the potential influence of the Lean Startup approach on the use of agile practices. The overall research questions that guide our study are:

RQ1: *Are software startups applying agile practices?*

RQ2: *Are software startups that adopt Lean Startup applying agile practices?*

To depict the state of the practice, we utilize the data collected in a large online survey conducted from September 2013 to September 2014. Based on the responses from 1526 surveyed software startups worldwide, we could obtain a good understanding of the state of the practice of agile practices applied in software startups.

The rest of the paper is organized as follows. Section 2 reviews the related work that has been conducted so far to understand the use of agile practices

in software startups. In Sect. 3, we explain how the online survey is utilized to answer the research questions. The findings are presented in Sect. 4 and further discussed in Sect. 5, together with the reflection on the limitations of and validity threats to the study. The paper ends with Sect. 6 in which potential future work is outlined.

2 Related Work

2.1 Agile Methods in Software Startups

The emergence of agile methods was a response to the inability of heavy-weight, waterfall-like development methodologies to allow software organizations to respond to change. Popular agile methods, such as Scrum and XP, have been adopted by both small and large companies worldwide over the years, rendering agile a mainstream software development approach [11]. At their core, agile methods focus on incremental and iterative development. The nimbleness and flexibility allowed by different agile practices, such as short iterations, continuous integration, etc., enable software organizations to address change effectively [12, 13]. The effective adoption and use of agile methods in established companies have been manifested in a growing body of agile research [14–16].

When the context is switched to software startups, the picture is less clear-cut. Some studies suggest in a general manner that agile methods are viable and suitable for software startups (e.g., [17, 18]). For example, Duc and Abrahamsson [9] find that four out of five startups they studied have adopted agile development processes. However, these studies do not specify clearly which particular agile method or agile practices have been used in software startups.

Other studies suggest a different picture. Coleman and O'Connor [5] argue that startups are creative and flexible in nature and are reluctant to introduce process which may hinder their natural attributes. They have very limited resources and typically wish to use these resources to support product development. Giardino et al. [18] observe that, to quickly validate the product in the market, software startups tend to use agile methods, but in an ad-hoc manner. Yau and Murphy [8] go further and contend that, given that the communication and cooperation dynamics in startups are very different from more established companies and the fact that the initial focus of a startup might be significantly different from its final objective, even the agile approach seems to impose too much rigidity and process on them. Without denying that agile methods offer clear benefits to startups over some of the more traditional methods, the authors question whether they are appropriate in tackling the problems faced by startups. Doubts are cast on the usefulness of agile practices including test-driven development, pair programming, user stories, velocity and backlogs [8].

2.2 Lean Startup and Agile Practices

The Lean startup approach is considered a variant to agile methods in software engineering literature [18]. It advocates the identification of the most risky parts

of a software business and the use of Minimum Viable Products (MVPs) to systematically test them and change the course of the development if needed. According to Ries [1], a MVP is “[the] version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort.” MVPs should be the main focus of both business and product development activities in software startups [9]. The strategic change is termed *pivot* in Lean Startup.

Even though the Lean Startup approach is seen as a recent advancement in agile community, and regarded by some as a more “extreme” agile approach than extreme programming (XP) or Scrum [19], difference between the two has been argued. Agile methods seem to be able to prescribe on how to develop a working software faster, but are unable to provide the answer to what product should be developed in the first place [20]. Although agile methods advocate to build software iteratively, they only work when problems are known to the stakeholders. Instead, startups typically are looking for right problems to solve and need to figure out who are their customers [2]. Lean Startup advocates startups to build products iteratively and get early feedback to test riskiest assumptions about their business models. The combined use of agile and Lean Startup seems a sensible approach for software startups.

The research conducted by Duc and Abrahamsson [9] is focused on different types of MVPs that software startups utilize and what are their main purposes. They argue that the adoption of MVP might be influenced by many contextual factors, and one most relevant factor is the product development methodology. They further suggest that the continuous integration—one of the agile practices—might be the impetus for the popular adoption of evolutionary prototypes and single-feature MVP in four of the five cases they studied.

However, Yau and Murphy [8] contend that certain agile practices may not be in consistency with the primary focus of software startups that adopt the Lean Startup approach. Quality is important for a software startup but cost and time may be larger deciding factors. A small scale startup that has not obtained much funding will probably have a short runway, and thus a limited amount of time and money. The priority in this case should be to create an MVP, which may lack in quality but is functional enough to show to investors. Terho et al. [10] also argue the need of balancing between quality and speed in creating MVPs, the intensified “developers dilemma” faced by software startups. As a consequence, the agile practices that are focused on quality of software, such as test-driven development and refactoring, may be compromised or even not taken on board.

3 Research Method

3.1 Survey Questions

This study utilizes a large online survey that was conducted between 2013 and 2014. The original survey explored various aspects of startups and covered a large set of questions. The authors had the opportunity to access the survey data and select the questions that were pertinent to the purpose of this study.

Table 1 shows the list of questions used in this study, as the result of the selection process. The questions are mainly divided into three categories:

- questions related to the demographic information of the respondents and the background information of the surveyed software startups;
- questions related to agile practices, which form the core category. We used the list of agile practices reported in the 10th annual agile report from VersionOne [11] as the commonly accepted agile practices. The original survey includes five questions relevant to agile practices: regular refactoring, test first, frequent release, agile planning, and daily standup meeting. All are close-ended questions. Four are ordinal and the one about daily stand-up meeting is binary. All of them require a single answer and are not mandatory.
- questions related to the Lean Startup approach, which allow us a more focused examination of the use of agile practices in software startups that adopted the Lean Startup approach. We identified three questions from the original survey which indicate whether a startup is following the Lean Startup approach or not. These questions reflect the key Lean Startup concepts: hypothesis-driven, MVP and pivot.

Table 1. Survey questions used in the study

<i>Background questions</i>	
About respondent	Select your gender How old are you? What is your motivation with this startup?
About startup	What kind of startup are you a part of? What is the total size of your team? How many founders are there on your team that own a significant piece of equity? What's the stage of your primary product? How many core features does your product have?
<i>Agile practice related questions</i>	
Regular Refactoring	How often do you refactor code?
Test First	When do you start writing tests?
Frequent Release	What is the frequency of your product release cycle?
Agile Planning	How far ahead do you plan your product development pipeline?
Daily Standup	Do you do daily stand up meetings?
<i>Lean Startup related questions</i>	
Hypothesis-driven	We identified the riskiest hypotheses about our business in order to test them first
MVP	We built minimally viable products to test our hypotheses
Pivot	How many pivots have you had?

3.2 Data Cleaning and Validation

To ensure the quality and validity of the survey data, we went through a careful data cleaning and validation process on the original dataset, which is described in detail in this section. The process was mainly automatized using R software package. Additionally, we have removed suspicious data entries manually.

To start with the data cleaning process, we set the threshold of 50 (out of a total of 278 original survey questions) as the minimum number of answered questions that a data entry should contain. All the rows with less than 50 answers were removed from the dataset. Afterwards, we merged rows if they were answered by the same person and for the same startup, because the survey collection application saved the data as a separate entry if the survey was interrupted and then restarted again. We also removed duplicate columns that might have been introduced during the data exporting process. We also fixed various obvious errors that may be attributed to the original survey design or data exporting process.

After this rudimentary step, we started automatized and manual data cleaning column by column (question by question). We removed all the rows where startup names were missing, to ensure that respondents have answered the questions by referring to specific startups. We also removed the rows with empty emails. We have decided to exclude from the sample the answers referring to the same startup but answered by different respondents because there was not a convincing rationale as to which answer to keep: the CEO's or the developer's. Each of them has its pros and cons. Fortunately there were not many duplicated startups. We also checked startup names, emails and websites and further removed the rows with suspicious values, for example, the answers that containing "none", "not", "test", "xyz", "untitled", etc. We then applied the regular expressions to all the columns that had a fixed set of values to further remove invalid answers. For example, if a question was Boolean, we ensured that only "0"s and "1"s were in the corresponding column. In the last step, we printed all the possible values for each closed question and ensured that only the valid answers were present in the dataset.

After the initial cleaning, we checked the validity of the data using a set of validation cases that we discovered based on a close inspection of all the survey questions. The validation cases detected a set of unrealistic, impossible, invalid combinations of answers which rendered certain data entries invalid, which in turn were removed from the dataset. All the validation cases we used are described in an online document that can be accessed at <https://figshare.com/s/08c35ec98fd85e827594>

The original dataset had 10171 entries. After applying the data cleaning and validation process, the final cleaned dataset has a sample size of 1526. By performing such strict data cleaning and validation steps, we may have removed some valid entries unintentionally. But removing some valid entries is a trade off that is worth making in order to obtain a clean dataset to conduct the data analysis.

3.3 Data Analysis

To answer the research questions posed in Sect. 1, we analyzed the data in two steps:

Step 1: To answer RQ1, firstly the structure of the five questions related to agile practices was analyzed using exploratory factor analysis. Two factors fit the model and the practices group in pairs: regular refactoring with test first, and frequent release with agile planning. Instead daily standup meeting does not show a significant correlation with any of the two factors. Therefore three dimensions can be defined to group the five agile practices: quality (regular refactoring and test first), speed (frequent release and agile planning), and communication (daily standup meeting). Next the internal consistency between the two items in the quality and speed dimensions was analyzed using Cronbach's alpha. However a low level of reliability estimates ($\alpha = 0.41$ and $\alpha = 0.50$ respectively) was obtained, which meant that the two items within each dimension were not suitable to aggregate. Therefore, the further analysis was conducted on each individual agile practice, rather than at the group level. To allow a sharper comparison, for each agile practice, we divided the startups into using the practice vs. not using it based on their answers to the question. In this way we converted the four agile questions that were categorical (ordinal) into binary. We examined the frequency of the use of five agile practices in the surveyed startups. Since software startups at different product development stages may adopt agile practices differently, we further investigated the difference using Chi-square. The hypothesis for each of the agile practices can be formulated as the following:

H_{a1} : There is significant difference in the use of [the agile practice] among software startups at different product development stages.

Step 2: The focus of this step was to analyze the use of agile practices in the software startups that adopted the Lean Startup approach, in order to answer RQ2. To identify lean software startups in the sample, we used the three questions related to the Lean Startup approach, as explained in Sect. 3.1. The software startups that answered "yes" to the first two questions and have pivoted at least once were considered adopting the Lean Startup approach therefore lean software startups. 229 out of 1526 are lean startups. The use of the five agile practices in these lean startups was compared to that in the rest of the whole sample, to understand if there was difference in agile practice use between the two sub samples. For this purpose again Chi-square tests were used. The hypothesis under the test regarding each agile practices can be formulated as the following:

H_{a2} : There is significant difference in the use of [the agile practice] between lean software startups and non lean software startups.

Since pivot is an important aspect of software startups, we also examined the number of pivots the surveyed startups made as part of Step 2 analysis.

The data analysis process was conducted using R software environment.

4 Results

The cleaned dataset contains information about 1526 software startups, provided by 1526 respondents who either founded or worked for these startups. Not surprisingly only a very small percentage (8%) are females in comparison to the much larger percent of males (76%, the remaining 16% didn't reveal gender information). The age of these respondents spread from 18 to 72 (based on 1219 cases that contain age information), with a mean of 34 and a median of 32 ($sd = 9.58$). A slight majority of the respondents (52.3%) have the age between 25 and 35. It is intriguing to understand what motivated the respondents to found or work for these startups. As expected the majority of answers reflect an entrepreneurial mindset: "Build a Great Product" covers the 52% of the motivations, followed by "Change the world" (29%). "Make a Good Living", "Get Rich" and "Create a quick flip" are motivations for only less than 20% of the respondents.

Regarding the types of these software startups, more than half of them (877) are working on web-based products. 264 software startups provide both web and mobile solutions. Mobile applications are the focus of 171 startups. Only 65 startups provide non web-based software solutions. The remaining 149 startups either work on products where software plays a less significant role or did not provide specific information regarding the types of their startups.

1461 software startups answered the question "What is the total size of your team?" with meaningful values. The distribution of the sample is skewed right significantly, with 81.2% of the software startup teams with less than 9 members. The mean of the team size is 7.23 and the median is 4 ($sd = 19.15$). When the number of founders is concerned, even though we could not obtain the direct data from the survey, we could infer from the question "how many founders are there on your team that own a significant piece of equity?" that most often an entrepreneurial team has two co-founders that have significant equity of the company, followed by 1-significant-founder and trio co-founder teams.

The distribution of the software startups across product development stages is shown in Fig. 1. It can be seen that it follows a normal distribution, with software startups that have functional products with limited users as most common, and those with mature products as the minority. A closer look at the number of core features that these products have reveals that the average number of the core features of a product is 5 ($mean = 5.2, median = 4, sd = 4.07$). 72% of the startups work on products that have 5 core features or less.

Product stage	Number of startups
Concept	121
In development	230
Working prototype	267
Functional product with limited users	721
Functional product with high grow	123
Mature product	60

Fig. 1. Startup distribution with respect to product stages

4.1 Agile Practices in Software Startups

Two agile practices, regular refactoring and test first, allow software startups to focus on the quality of their products. 1240 startups responded to the question related to regular refactoring, and 1273 to test first. As shown in Fig. 2, regarding refactoring, slightly less than 45% of the startups do care about the quality and refactor the code every few weeks or even once a week. However, a bit more than one fourth of those rarely or never do refactoring. If refactoring “once a week” and “every few weeks” are considered regular therefore an agile practice (blue bars in Fig. 2), the other options indicate that regular refactoring is not practiced in the startups. It can be concluded that a slight majority of the startups surveyed are not doing regular refactoring.

Regular refactoring	Percentage of startups
Once a week	15.2%
Every few weeks	29.6%
Every few months	18.7%
A few times a year	11.2%
Rarely/never	25.3%

Fig. 2. Startup distribution with respect to the frequency of code refactoring (Color figure online)

Similar results are shown in the test first practice. It is evident from Fig. 3 that around 32% of them are writing tests as soon as they write features, therefore practicing test first (blue bar). However, again one fourth of the startups never write tests. Among the other options, “as soon as we know we’re going to keep a feature” indicates clearly the test first practice is not used. Even though we could not interpret properly the options “as soon as we reach a legal agreement with a customer” and “other” due to a lack of access to the original survey design, we could still conclude that the majority of the startups surveyed do not adopt the test first practice.

Test first	Percentage of startups
As soon as we write a feature	32.3%
As soon as we reach legal agreement with a customer	3.7%
As soon as we know we are going to keep a feature	26.2%
Never	25.1%
Other	12.7%

Fig. 3. Startup distribution with respect to the frequency of code testing (Color figure online)

Agile planning and frequent release are the two practices that allow software teams to be able to collect feedback on their products and adjust their development speed accordingly. 1391 startups responded with their release frequencies and 1290 indicated how far ahead they planned their product development

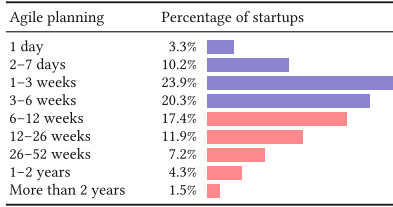


Fig. 4. Startup distribution with respect to agile planning (Color figure online)

pipelines. Regarding planning, Fig. 4 shows that most often the software startups plan ahead for 1 to 3 weeks (about 24%), more than 10% plan for 2 to 7 days, and about 3% are doing daily planning. Only less than 6% put up a yearly or longer-term plan. In total, more than 57% of the startups plan in an agile manner in terms of the time frame covered by the planning (shown by the blue bars. We used 30-day sprint to draw the division). Agile planning should be for 3 to 6 weeks (30 working days) or shorter.

As shown in Fig. 5, the most common (about 21%) release frequency used by these software startups is every 2 to 3 weeks, followed by every 1–3 months (about 19%). It is interesting to see that more than 13% of the startups are practicing continuous delivery and release product once per day or even multiple times per day. However, more than 15% other startups have really low release frequency (every 3–6 months or even more than 6 months), which is worrying given the fact that they are software startups and moving fast is not an option but a must for many of them. The bars in Fig. 5 are divided into two groups: those with release frequency of 2–3 weeks or less (blue bars) therefore indicating frequent release (again the 30-day sprint length was used as the division line), and those indicating low release frequency (taking more than one sprint to release a new version). It can be seen that more than 64% of the startups do frequently release their products.

Daily standup meeting is an agile ceremony used to facilitate communication among software development teams and organizations. Among the 1286 software startups that answered the question, more than 70% are not using the practice, in contrast to about 30% that said “yes” to the question.

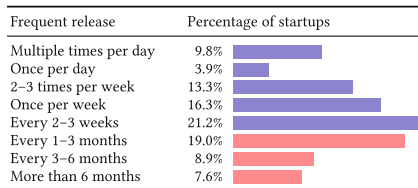


Fig. 5. Startup distribution with respect to frequency of product releases (Color figure online)

Table 2. The use of agile practices in software startups across product stages

Product development stage	Regular refactoring		Test first		Frequent release		Agile planning		Daily standup meeting	
	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No
Concept	49	47	41	41	2	0	63	36	23	76
In development	93	93	74	97	144	80	118	78	53	146
Working prototype	111	107	87	119	158	108	138	91	59	170
Functional product with limited users	246	337	190	323	483	230	350	257	198	403
Functional product with high growth	40	63	42	51	79	44	53	51	32	72
Mature product	16	35	24	19	29	31	20	32	16	35

Table 2 shows the use of the five agile practices by the software startups across different product development stages. As explained in Sect. 3.3, the use of the agile practices are simplified into “yes”/“no” Boolean options, to allow a sharper comparison. Table 2 does show that for each agile practice, the percentage of software startups using it varies across the product development stages. However, there is no discernible pattern in the variance of the percentages.

To test H_{a1} , Chi-square tests were applied. A pre-examination excluded frequent release from the test since the assumptions requested to run Chi-square test were not met. We run the tests on the cleaned sample ($n = 1526$). Since the data entries that have empty answers to each agile practice and/or product development stage were removed, each test has a different sample size (as shown in Table 3, Column 2). The test results show that regular refactoring and agile planning are linked to the development stages (the respective H_{a1} is supported). Instead, H_{a1} regarding test first and daily standup meeting cannot be supported.

4.2 Agile Practices in Lean Software Startups

Regarding the individual responses to the three Lean Startup questions from the whole sample, 489 out of the 1526 replied with a definitive “yes” to the statement

Table 3. Agile practices across product stages—Chi-square test results

Practice	n	Chi-square	Degrees of freedom	p -value	Result
Regular refactoring	1237	13.638	5	0.01808	H_{a1} supported
Test first	1108	11.06	5	0.05021	H_{a1} rejected
Agile planning	1287	12.365	5	0.030121	H_{a1} supported
Daily standup meeting	1283	7.736	5	0.1714	H_{a1} rejected

Table 4. Pivoting in lean startups across product stages

Product stage	No. of lean startups	Mean of number of pivots
Concept	3	2.0
In development	40	2.1
Working prototype	57	2.4
Functional product with limited users	107	2.0
Functional product with high growth	17	2.4
Mature product	5	2.6

“We identified the riskiest hypotheses about our business in order to test them first”, and 55% claimed that “We built minimally viable products to test our hypotheses”. It is interesting to explore the pivoting behavior of these startups in terms of the number of pivots they have made. 1440 out of 1526 gave valid answers to the number of pivots. The mean is 1.528 and median is 1 (sd = 2.06), in a range from 0 to 30 pivots.

229 out of the 1526 software startups are considered following the Lean Startup approach based on the selection criteria specified in Sect. 3.3. When looking closely at the pivoting in this subset, the number of total pivots the surveyed startups experienced ranges from 1 to 15, with the mean equal to 2.153 and the median to 2 (sd = 1.73). From the perspective of product development stages, we can see that, as shown in Table 4, the mean of the number of total pivots of startups at different stages ranges from 2 to 2.6. The lean startups that progressed to the stages of having functional or mature products in total have not pivoted more than those at the early product development stages.

Table 5 shows the use of the five agile practices in lean startups in comparison to that in the rest of the sample. It can be seen that there is a higher percentage of lean startups using each of the agile practices for all the five agile practices.

To test H_{a2} , we used the Chi-square test on the two groups: lean startups vs. non-lean startups. The results are shown in Table 6. The difference between

Table 5. The use of agile practices in lean startups vs. non lean startups

Agile practice	Lean startup subset		Non lean startup subset	
	Yes	No	Yes	No
Regular refactoring	99	99	456	586
Test first	86	86	372	567
Frequent release	164	61	733	433
Agile planning	119	83	626	462
Daily standup meeting	76	124	306	780

Table 6. Agile practices in lean vs. non-lean startups—Chi-square test results

Practice	n	Chi-square	Degrees of freedom	p -value	Result
Regular refactoring	1240	2.3723	1	0.1235	H_{a2} rejected
Test first	1111	6.0471	1	0.0139	H_{a2} supported
Agile planning	1290	0.0815	1	0.7752	H_{a2} rejected
Frequent release	1391	7.8438	1	0.0051	H_{a2} supported
Daily standup meeting	1286	7.3417	1	0.0067	H_{a2} supported

the two groups is not significant in terms of the use of regular refactoring and agile planning. Instead, the percentage of lean startups using test first, frequent release or daily standup meeting is significantly higher than that of non-lean startups.

5 Discussion

So are software startups using agile practices? The results of our study reveal that a majority of software startups do not use quality related agile practices, such as regular refactoring and test first. It reflects the major concern expressed in the literature that quality has a low priority and technical debt is accumulated in software startups, especially at their early stages. When the agile practices regarding the speed of development are concerned, our study shows that a large majority of software startups do move fast by adopting frequent releases and short-term agile planning. This is in line with the literature that emphasizes that speed matters significantly to software startups [7]. However, the under use of quality related agile practices in comparison to speed related practices is not unique to software startups. The same pattern has been manifested in the surveys of agile and lean adoption in software organizations in general. For example, in the 10th annual agile survey conducted by VersionOne (based on 3,880 completed responses) [11], it is shown that speed related practices (e.g., short iterations, iteration planning, release planning) are employed more often in the surveyed organizations than quality related practices (such as unit testing, refactoring, test-driven development). A smaller scale academic survey on agile and lean usage in Finnish software industry with 408 responses demonstrates the same tendency [21]. It seems that, in terms of balancing speed and quality concerns, software startups are not so different from the general population of software organizations. Agile practices related to speed are more often used by both software startups and established companies alike.

In contrast, our findings regarding daily standup meeting indicate that this well-known agile practice is not used in software startups to the same extent as in established software organizations. According to the VersionOne survey [11], daily standup meeting is the most popular agile practice among the surveyed

organizations, with an adoption rate of 83%. Its popularity is echoed in the academic survey too [21]. In our survey instead, daily standup meeting is the least frequently used practice among the five agile practices studied. Only about 30% of the software startups use this practice. One explanation of such different could be that daily standup meeting is a typical agile ceremony used by software development teams and organizations to facilitate communication. Because most startup teams have very small sizes (as described in Sect. 4), informal communication happens frequently, which renders formal communication practices less necessary. Yau and Murphy [8] offer similar arguments. They contend that, in small scale startups with only a few members, many problems that agile methods set out to solve do not exist, e.g., the communication issue.

In this study we further examined the use of agile practices by software startups at different stages of product development. The results of the hypothesis testing (H_{a1}) show that the use of agile practices including regular refactoring and agile planning does vary across the product development stages. Instead, the use of test first and daily standup meeting is not significantly associated with the stages. We cannot draw any conclusion regarding frequent release. This finding provides partial support to the claim in the literature that not all software engineering practices are usable or beneficial in different stages of startups [22]. It is an interesting direction to investigate which software engineering practices are most useful and beneficial to which stages of startups.

Another specific angle investigated in our study is the use of agile practices by software startups that adopted the Lean Startup approach. Some studies have expressed the concerns that startups adopting the Lean Startup approach have to sacrifice certain agile practices or product quality due to limited funding and short runway in order to move fast and test business hypotheses with MVPs [8, 10]. However, the findings reported in Sect. 4.2 do not substantiate these concerns. On the contrary, they reveal that lean software startups tend to use agile practices more than the rest of the startups surveyed. Especially in terms of test first, frequent release and daily standup meeting, significantly higher portions of lean startups practice them. With these practices that address both needs of quality and speed, lean software startups may be in a good position to manage the “developers dilemma” [10], better at balancing between quality and speed to achieve fast product iteration.

Even though not a main focus of this study, it is worth noting the somehow surprising finding regarding the number of pivots made by lean startups across different product development stages. Pivot is considered a key component of the Lean Startup approach, an action that startups are encouraged to take based on the validated learning they obtain through testing risky business assumptions early and often [1]. Therefore, one would expect that the total number of pivots increases as startups progress along the development stages and pivot continuously. However, the result regarding pivoting reported in Sect. 4.2 does not conform to this expectation. Further investigation is needed to understand the pivoting in software startups.

Lastly, the results reported in this paper need to be viewed in the lights of the limitations of and validity threats to the study. The lack of access to the original survey design and no control to the quality of collected data pose the biggest limitation to our study, constraining the types of analysis that can be conducted and consequently the results that can be obtained. For these reasons, we went to great lengths to clean and validate the data to ensure its quality. Another limitation is due to the fact that there are a very limited number of questions in the original survey that can be associated with agile methods and practices with an acceptable level of confidence. At the end only five agile practices were brought into the study. In addition, each agile practice had only one corresponding question (item), so the risk of not obtaining valid data was increased due to the lack of multiple items to probe the same practice. These concerns pose a potential threat to the construct validity of the study. Instead, the external validity is ensured by the size and random nature of the sample. Therefore the findings of this study can be generalized to a general population of software startups.

6 Conclusion

In the past years agile methods have become main-stream software development approaches in established companies, small or large. They are considered natural choices for software startups too, since startups operate under various uncertainties and the demand on their ability to deal with change is high. Meanwhile software startups have to focus on business development as well as product development. Lean Startup is the approach that an increasing number of startups adopt to test the riskiest business assumptions in their business models. This study provided a better understanding of the state of agile practices in software startups, with a particular focus on lean startups. Based on a large survey of 1526 software startups, we found out that different agile practices are used to different extents, depending on the focus of the practices. Speed related agile practices are used to a greater extent in comparison to quality related practices. Communication practices represented by daily standup meeting is least used. In addition, unlike what is speculated in the literature, software startups who adopt the Lean Startup approach do not sacrifice quality for speed more than other startups do. Our study is the first step towards more in-depth understanding of how software startups can better use agile practices and eventually benefit from them.

In our current study we could not identify any questions specific to lean practices, such as kanban, from the original survey questions. Future work can investigate how lean practices are used in software startups. Meanwhile, “doing agile”, using agile practices, does not ensure software startups of “being agile”, being able to respond to change and uncertainty. This study was focused on “doing agile”. Future work can assess the agility of software startups, and establish the link between “doing” and “being” agile to startup success. It would be also effort worth spent to design a new survey that is focused on investigating the adoption of agile and lean methods as well as Lean Startup in software startups.

Acknowledgement. Thanks a lot to Carmine Giardino who shared the original survey data with us.

References

1. Ries, E.: *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business, New York (2011)
2. Blank, S.G.: *The Four Steps to the Epiphany: Successful Strategies for Products that Win*. Cafepress.com, Foster City (2005)
3. Sutton, S.M.: The role of process in a software start-up. *IEEE Softw.* **17**, 33–39 (2000)
4. Unterkalmsteiner, M., Abrahamsson, P., Wang, X., Nguyen-Duc, A., Shah, S., Bajwa, S.S., Baltés, G.H., Conboy, K., Cullina, E., Dennehy, D., et al.: Software startups—a research agenda. *e-Informatica Softw. Eng. J.* **10**(1), 89–123 (2016)
5. Coleman, G., O'Connor, R.V.: An investigation into software development process formation in software start-ups. *J. Enterp. Inf. Manag.* **21**(6), 633–648 (2008)
6. Thomas, S.: Done is better than perfect: how to beat perfectionism paralysis (2016). <http://engageme.online/done-is-better-than-perfect-how-to-beat-perfectionism-paralysis/>
7. Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: the greenfield startup model. *IEEE Trans. Softw. Eng.* **42**(6), 585–604 (2016)
8. Yau, A., Murphy, C.: Is a rigorous agile methodology the best development strategy for small scale tech startups? Technical report (CIS), Paper980, p. 9 (2013)
9. Duc, A.N., Abrahamsson, P.: Minimum viable product or multiple facet product? The role of MVP in software startups. In: *Agile Processes, in Software Engineering, and Extreme Programming*, vol. 251, pp. 118–130 (2016)
10. Terho, H., Suonsyrjä, S., Systä, K.: The developers dilemma: perfect product development or fast business validation? In: Abrahamsson, P., Jedlitschka, A., Nguyen Duc, A., Felderer, M., Amasaki, S., Mikkonen, T. (eds.) *PROFES 2016. LNCS*, vol. 10027, pp. 571–579. Springer, Cham (2016). doi:[10.1007/978-3-319-49094-6_42](https://doi.org/10.1007/978-3-319-49094-6_42)
11. VersionOne: *The 10th Annual State of Agile Report*. Technical report (2016)
12. Highsmith, J., Cockburn, A.: Agile software development: the business of innovation. *Computer* **34**(9), 120–127 (2001)
13. Beck, K., Andres, C.: *Extreme Programming Explained: Embrace Change*, 2nd edn. Addison-Wesley Professional, Boston (2004)
14. Dybå, T., Dingsøyr, T.: Empirical studies of agile software development: a systematic review. *Inf. Softw. Technol.* **50**(9), 833–859 (2008)
15. Abrahamsson, P., Conboy, K., Wang, X.: “lots done, more to do”: the current state of agile systems development research. *Eur. J. Inf. Syst.* **18**, 281–284 (2009)
16. Dingsøyr, T., Nerur, S., Balijepally, V., Moe, N.B.: A decade of agile methodologies: towards explaining agile software development. *J. Syst. Softw.* **85**(6), 1213–1221 (2012)
17. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: a systematic mapping study. *Inf. Softw. Technol.* **56**(10), 1200–1218 (2014)
18. Giardino, C., Unterkalmsteiner, M., Paternoster, N., Gorschek, T., Abrahamsson, P.: What do we know about software development in startups? *IEEE Softw.* **31**(5), 28–32 (2014)

19. Gilb, T., Gilb, K.: “Lean Startup” - the most extreme agile method by far. *Agile Rec.* (9), 53–54 (2012)
20. Bosch, J., Holmström Olsson, H., Björk, J., Ljungblad, J.: The early stage software startup development model: a framework for operationalizing lean principles in software startups. In: Fitzgerald, B., Conboy, K., Power, K., Valerdi, R., Morgan, L., Stol, K.-J. (eds.) *LESS 2013. LNBIP*, vol. 167, pp. 1–15. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-44930-7_1](https://doi.org/10.1007/978-3-642-44930-7_1)
21. Rodríguez, P., Markkula, J., Oivo, M., Turula, K.: Survey on agile and lean usage in finish software industry. In: *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM 2012*, p. 139 (2012)
22. Eloranta, V.P.: Towards a pattern language for software start-ups. In: *19th European Conference on Pattern Languages of Programs*, pp. 1–11 (2014)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

