An empirical study on software engineering and software startups: findings from cases in an innovation ecosystem

Leandro Pompermaier*, Rafael Chanin†, Afonso Sales‡, Kellen Fraga § and Rafael Prikladnicki¶

* † ‡ ¶Computer Science Department
§Business School
Pontifical Catholic University of Rio Grande do Sul (PUCRS)
Porto Alegre, RS, Brazil
*leandro.pompermaier@pucrs.br, †rafael.chanin@pucrs.br, ‡afonso.sales@pucrs.br,
§kellen.silva@pucrs.br, ¶rafaelp@pucrs.br

Abstract—**Context:Software startups are increasingly seen as opportunities in a varieties of areas; from simple applications, as personal controls, to complex solutions, involving artificial intelligence or using big data. The process of developing these solutions happens quickly and in abbreviated form. Thus, software engineering should be adapted to better understand this world of software startups in a way to assist them in their customer discovery challenges and in the search for scalable business models. Main Goal: Understand the factors that positively and negatively influence the software development of software startups and present solutions found in an empirical study on startups located in an innovation ecosystem Methodology: We conducted interviews and observations as data collection techniques in eight software startups located at a Tech Park.**

Keywords*—*Software Startups; Software Engineering.

I. Introduction

The technological evolution presented in the last years presents a significant challenge for the companies in general since they concentrate efforts to find new and more efficient ways of conducting their businesses [8]. There is a growing market for systems that operate at low cost, high quality and have a short development cycle. It suggests that the development approach using software engineering practices start on the need to achieve high quality and productivity.

In software development, some of the key factors affecting quality and productivity include people and procedures as well as technologies being employed to accomplish the intended activity [2]. Currently, there is a growing number of startups that develop innovative solutions. Startup could be defined as a human institution designed to deliver a new product or service under conditions of extreme uncertainty [10].We can define a subset of startups that have their software-based solutions as software startups or digital startups. These are considered to be newly created companies with no operational history and are incredibly fast in the production of cutting-edge technologies [9].

Innovation environments, such as universities, play a fundamental role in technological innovation and socio-economic growth in a region or country. According to Etzkowitz [6], the transformations that universities have undergone in recent years, the so-called University revolutions, have unified the three top strands of these institutions: teaching, research and economic and social development of the country. Business incubators are mechanisms of innovation and strategic process in the economic development of countries, states, cities and companies. The aim of incubators is to produce successful enterprises, which is why incubators help ventures to survive and grow during their early stages.

Furthermore, software startups are increasingly obsessed with delivering a software product in an extremely short time besides validating the solution directly with the end user. The use of lean software development methodology and the business models have become popular in software startups, especially in the design of the minimum viable product. Thereby, some problems can be identified by adopting (or not) software engineering practices during the early stages of the startup life cycle.

So the answer we are looking for is related to the further research question: *How are software engineering practices being applied in the software product development of software startups?*

The objective of this article is to present an empirical study on the development of MVPs in software startups located in innovation environments in southern Brazil and providing the answers to the research question. This article is organized as follows: in Section 2, the central concepts that guide this work are presented. In Section 3, the method used is described, and in Section 4 we present the results and the insights of this empirical research. Finally, in Section 5, we state our conclusions.

II. Background

A. Software Startups

Software startups are newly created companies that do not yet have a history of the technological world and operate in a

DOI reference number: 10.18293/SEKE2017-115

highly volatile environment with innovative products[4]. They are also called digital startups and find challenges in their insertion in the market. For Steve Blank [1], a software startup or simply startup is a temporary organization looking for a repeatable and scalable business model, while Ries [10] defines it as an institution designed to create new products and services under extreme conditions uncertainty. Startups should have a strategy to develop their prototype and then the product, to then secure a market base for their new products and services, but they have limited access to resources[4]. A startup cannot be confused with a small company. While an established (even small) company has a validated business model, startups are in the pursuit of a business model. Startups have some recurring features [9]:

- Little experience;
- Limitation of resources;
- Various influences;
- Dynamic technology and markets.

The development of a software startup has a different lifecycle than a traditional software company, that is, enterprises that use traditional software development methodologies [19]. Steve Blank [1] defined a four-step process for the initial development of a startup: Customer Discovery, Customer Validation, Demand Generation, and Enterprise Structuring. According to Bosch [2], software startups go through several stages before they reach maturity levels, that is, they have to go through several steps until they can create a business model that works and is feasible.

Usually, decisions related to the development of the product of a startup, such as the use of software engineering practices or not, are part of the executive team of the startup [11]. However, the initial gain achieved regarding flexibility and speed is counterbalanced by the need to restructure the product where the business begins to grow [7].

B. Software Development

All aspects of software production, from early stages to system maintenance, involve specifying, developing, managing and evolving software systems [11]. This procedure is the work of Software Engineering (SE), which arose to solve problems of software systems, aiming to support the development of software using processes, methods, techniques and tools [11]. Techniques such as elaboration and implementation of a computer system consist of the software development process, one of the SE subareas. This subarea aims to transform users needs into a software product capable of solving the user's problems.

The prescriptive models (cascade, prototyping, incremental, among others) are models that prescribe how a new software system should be developed. Prescriptive models are used as guidelines or structures to organize and structure how software development activities should be performed and in what order. Regardless of the process, the following activities are critical to software engineering [11]:

- Software Specification: Definition of functionalities and constraints;
- Design and implementation of software: Production of software complying with requirements;
- Software Validation: Checks if the software meets what the customer wants;
- Evolution of software: The software must evolve to meet the changes requested by the client.

III. Method

The methodology used in this research was based on a Eisenhardt study [5]. In summary, the method steps include a definition of the research, selection of cases, formulation of research instruments and protocols, data collection, analysis by comparison within and between cases, formulation of hypotheses, comparison with literature and theoretical saturation when possible.

A. Getting Started

Initially, we made the definition of Research Questions, which according to Eisenhardt [5] is fundamental to focus on the construction of theory based on the case study. For this case study, we selected startups that have the software base and are located in an innovation environment (business incubator or science and technology park).

Also, we defined factors related to this sample, based on aspects such as:

- Level of maturity: in this case we consider companies that had already developed their product, regardless of whether this product was accepted by the market (market fit) or not;
- Technical partner: important in a software startup is the existence of a partner that understands the technical part and leads the development of the product;
- Interactions with the ecosystem: we consider it important for the object of this research that the number of interactions with the ecosystem of innovation was greater than 1.

B. Selecting Participants

In undertaking case studies, there is the selection of cases. The selection of an appropriate population controls extraneous variation and helps in the definition of limits in generalizing the findings. In case studies related to population, one must be strategic to offer clarified domains of the findings [5].

Using the factors presented previously, our search was for companies that were within the context of innovation, that is, that were inhabiting environments or business incubator or science and technology park. This decision was made because we understood that the analysis of the research question itself could be enriched if we could analyze startups that received the same development opportunities for their development. And in this case, companies in these environments show these patterns. Thus, we chose companies that have their operational bases either in the RAIAR Incubator or the TECNOPUC of PUCRS, Brazil.

To know, RAIAR is an incubator of companies that operates in the lean line of development of software startups and is located in the scientific and technological park, TECNOPUC.

C. Collecting Data

We used more than one source of data and method of collection to increase consistency and reliability: interviews and field observations.

Semi-structured interviews were conducted with 8 companies selected according to what has been previously specified. The interviews script was created with open questions to understand better the relationship of these startups with the ecosystem, as well as the story behind the product they offer to the market today such as the construction of the MVP, the decisions made, or development problems and how were solved which. These interviews were directed to the founders of the digital startups, and this choice was made so that we had the data necessary to understand how the business started and what technical decisions were made to change or not the development of the software product.

Also, an observation made during the incubation process performed by the team responsible for the companies was considered in this research. These observations were based on the structure of the technical team, difficulties faced and reported during the incubation process, such as problems with requirements, verification, system validation, and software architecture.

D. Analyzing Data

A striking feature of any research to build theory from case studies is data analysis, that can be achieved through notes [5]. In this research, we searched for patterns in the answers and categorize them according to the relevance of the proposed theme - software engineering. In this way, we can get an overview of how software engineering is being perceived and used by software startups that are in a technology park.

IV. Results

In this section, we present the achieved results from the field study with interviews, as well as some findings related to the use of software engineering during the development of MVP of software startups.

Among the companies studied the average interaction with the ecosystem was 2.5 years and during this time the majority (62.5%) did not have MVP, that is, software startups began their activities in the ecosystem with the first objective: to develop the minimum viable product. Another important point is the fact that the majority (87.5%) had the technical team (programmers) which shows the control in the software development process.

We could see that among all the activities related to software engineering, some were more evident in the problems presented by the startups during and after the MVP development. According to the interview responses and the areas of knowledge defined by SWEBOK [3], software requirements, software structure and architecture, and software testing are the critical points that will be discussed from now on.

A. Software Requirement

During software development, it is common to have changes related to user requirements. Typically changes occur due to the evolution of the business to which the software is linked and the cost to change a requirement upon software developed is considered too high. However, analyzing the life cycle of a startup (Section II.A) often the customer is unknown to the startup and the requirement management is compromised by this ”lack of certainty.”

Throughout the research, it was found that a good part of the interviewees (62.5%) use a pseudo agile method to manage the requirements of the MVP. That is, a visual management tool (kanban) and some Scrum practices are used. Basically, *post-its* are stuck on the boards indicating what should be done (new requirements), what the team is working on and what has already been developed. Furthermore, prioritization is a common practice among startups.

Asking the startups more about requirements management, 100% indicated that many requirements are not managed and/or documented, being passed on verbally either by the key user or the startup leader. When questioned about the reason for using some practices and not others, all of them indicated that the time they have should be used in product coding and avoiding the development process to be bureaucratic.

Our first finding comes up from the situation described before.

Finding 1 *Adoption of requirements management techniques lead to a significant increase in development time.*

A second finding can be defined as follows:

Finding 2 - *Non-documentation and/or requirement management does not interfere with the quality of the MVP developed.*

B. Software Structure and Architecture

Since most of the entrepreneurs in each company have a technical background, at several moments in the interviews, it was stated that there was the need for an improvement of the software structure and architecture. However, much of the development was carried out without adequate planning and without the use of structures and coding standards that allow the growth of the system (either in the number of functionalities or the number of users).

Table I presents the problems with software structure and architecture faced by the surveyed startups.

The following finding related to the software structure and architecture can be formulated:

Finding 3 - *A poor definition or non-definition of a software structure and architecture in the initial phase of a software startup exponentially increases the company's technical debt.*

C. Software Testing

Despite these numbers, the technical team in its entirety did not use software testing techniques in the construction of the first version of the system, leaving to the end user this responsibility. In the following versions of the system, this scenario changed considerably, 75% used some software testing technique.

TABLE I. PROBLEMS WITH SOFTWARE STRUCTURE AND ARCHITECTURE

Problem	Description
Coding language & associated IDE	The choice of programming language may affect future extensions of the software and integrations with existing solutions in the market.
Database	Need to choose a database that has an active user community, allowing a search for solutions to problems encountered while using MVP.
Web host	Web hosting is critical for the solutions developed by software startups. It needs to be an affordable and secure framework.
Code Deployment tool	Always making software changes and the associated implementations that come with it need to be controlled to increase team productivity and quality.
Security approach	Usually indicated as an important item by the startup team but ignored during development (using the security solutions offered by the web host).

In Fig. 1, we can see all different ways that these startups had been performing software testing in their MVP life cycle.

- Using a pilot client: It is considered here the system test performed by a group of end users, prior to deployment, to provide feedback to the product development team.
- Unit tests: a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.
- Functional ad-hoc tests: software testing performed without planning and documentation.
- Specialist tester: software test performed by a business specialist associated with the startup.

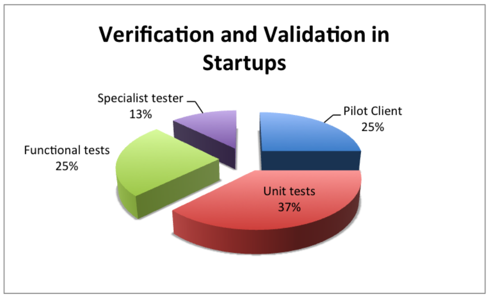

Fig. 1. Software testing in Startups

The way that the startups are performing software testing activities during their MVP life cicle leads to a fourth finding:

Finding 4 - *Structuring or formalizing a software testing step will increase the market acceptance of the requirements developed.*

V. Conclusion

In this paper we presented the results from an empirical study on the development of MVPs in software startups located in innovation environments in southern Brazil, mainly analyzing the problems encountered during the software development phases.

Based on the sample of startups studied, we verified that requirements, software structure and architecture, and software testing are critical and should be more detailed. The four findings defined in this paper should be better studied in order to find insights that can improve the development process used by startups.

The results showed in this paper provides an initial step for understanding what happens in the early phases of any software startup, mainly related to software engineering activities. A few limitations must be taken into consideration in this study. First, the sample size was small. Second, interviews were conducted only with startup founders. As future work we intend to consider a bigger sample of startups as well as a more diverse set of interviewees.

Acknowledgment

This work was partially supported by grants from BEPiD/PUCRS (Brazilian Education Program for iOS Development).

References

[1] Steve Blank. *The startup owner's manual: The step-by-step guide for building a great company*. BookBaby, 2012.

[2] Jan Bosch. Speed, data, and ecosystems: the future of software engineering. *IEEE Software*, 33(1):82–88, 2016.

[3] Pierre Bourque, Richard E Fairley, et al. *Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0*. IEEE Computer Society Press, 2014.

[4] Henry Edison, Xiaofeng Wang, and Pekka Abrahamsson. Lean startup: why large software companies should care. In *Scientific Workshop Proceedings of the XP2015*, page 2. ACM, 2015.

[5] Kathleen M Eisenhardt. Building theories from case study research. *Academy of management review*, 14(4):532–550, 1989.

[6] Henry Etzkowitz. The evolution of the entrepreneurial university. *International Journal of Technology and Globalisation*, 1(1):64–77, 2004.

[7] Carmine Giardino, Xiaofeng Wang, and Pekka Abrahamsson. Why early-stage software startups fail: a behavioral framework. In *International Conference of Software Business*, pages 27–41. Springer, 2014.

[8] Per Lenberg, Emil Alégroth, Robert Feldt, and Lars Göran Wallgren Tengberg. An initial analysis of differences in software engineers' attitudes towards organizational change. In *Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering*, pages 1–7. ACM, 2016.

[9] Nicolò Paternoster, Carmine Giardino, Michael Unterkalmsteiner, Tony Gorschek, and Pekka Abrahamsson. Software development in startup companies: A systematic mapping study. *Information and Software Technology*, 56(10):1200–1218, 2014.

[10] Eric Ries. *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Crown Business, 2011.

[11] Ian Sommerville. *Software Engineering*. Addison-Wesley Publishing Company, USA, 9th edition, 2010.