Challenge Based Startup Learning: A Framework to Teach Software Startup

Rafael Chanin
PUCRS, School of Technology
Av. Ipiranga 6681, Porto Alegre, Brazil
rafael.chanin@pucrs.br

Afonso Sales
PUCRS, School of Technology
Av. Ipiranga 6681, Porto Alegre, Brazil
afonso.sales@pucrs.br

Leandro Pompermaier
PUCRS, School of Technology
Av. Ipiranga 6681, Porto Alegre, Brazil
leandro.pompermaier@pucrs.br

Rafael Prikladnicki
PUCRS, School of Technology
Av. Ipiranga 6681, Porto Alegre, Brazil
rafaelp@pucrs.br

ABSTRACT

The advances in technology have enabled people to develop services and products that can reach millions of people around the world. Facebook and Google are examples of successful startups. However, most startups fail in their first years of life. From an education standpoint, even though universities are adapting their curriculums in order to embrace startup education content, the challenge still remains: how can we provide real world experiences for students to build relevant startups? In order to fill this gap, this paper introduces the Challenge Based Startup Learning, a framework that combines the Challenge Based Learning methodology with Lean Startup and Customer Development concepts, supported by software development techniques. We have applied and evaluated the framework in an undergraduate digital entrepreneurship course. Our preliminary results indicate that students not only understood what it takes to run a real startup, but they also felt engaged and empowered by delivering a useful and meaningful software.

CCS CONCEPTS

• **Social and professional topics** → **Software engineering education**; • **Information systems** → *Mobile information processing systems*;

KEYWORDS

Startup Education, Software Engineering Education, Entrepreneurship Education, Challenge Based Learning.

ACM Reference Format:
Rafael Chanin, Afonso Sales, Leandro Pompermaier, and Rafael Prikladnicki. 2018. Challenge Based Startup Learning: A Framework to Teach Software Startup. In *Proceedings of 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'18).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3197091.3197122

ITiCSE'18, July 2–4, 2018, Larnaca, Cyprus
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5707-4/18/07…$15.00
https://doi.org/10.1145/3197091.3197122

1 INTRODUCTION

Most universities across the world have already recognized the importance of providing entrepreneurial skills to engineering students [23]. Being technically competent is a must, but it is not enough. Knowing how to develop, market, and sell products and services is essencial to survive in the reality we live in. Several institutions are already providing programs and courses focused on entrepreneurship in order to fulfill this need [21].

Moreover, the advancement of new technologies along with the internet has been changing the way people interact. New startups are created every day; some of them are known for reaching billions of people across the globe [10]. Most startups follow the *lean startup* methodology, proposed by Eric Ries [24]. By combining customer interaction with short software development cycles, this methodology maximizes learning and minimizes risks. Sadly, most startups do not succeed [10]. Even though there are many factors that could lead to the failure of a startup, bad software engineering practices is pointed out as a key reason [7, 10, 14].

Companies and the academic community are aware about this issue. Several publications concerning software development processes in the context of a startup can be found in the most important database sources [10–12, 18, 22]. However, there are not so many studies focused on how software startup processes are taught to students in an academic environment. Since startups deal with real world challenges, problem-solving approaches could be effective to help students in understanding software startup processes. Challenge Based Learning (CBL) [20], for instance, was already combined with the Scrum framework [27] in order to teach mobile application development [25].

In this sense, the goal of this paper is to present a framework to educate students on software startups that combines CBL, lean startup, customer development and software development techniques. We call it *Challenge Based Startup Learning* (CBSL).

The remainder of the paper is structured as follows. Section 2 introduces the related concepts. Section 3 presents a study that had a significant impact on the proposed model. Section 4 describes the proposed framework. The application of the framework as well as our preliminary results are shown in Section 5. Finally, we conclude this paper in Section 6 exploring our final remarks.

2 BACKGROUND

2.1 Challenge Based Learning

Challenge Based Learning (CBL) [20] is a learning framework based on solving real world challenges. The framework was developed in a partnership between *Apple Inc.* and educators [19], and it has been applied in both professional and educational settings. From an education standpoint, students engage more in the learning process since they can clearly see the correlation between content and real life problems [17]. In addition, Johnson and Adams [13] have demonstrated that the use of active learning methodologies improves students' learning when compared to traditional methods. It is worth mentioning that the engagement and the soft skills acquired during the learning process is also viewed as a big asset for all stakeholders involved.

One important aspect of this process is that there is not hierarchy between students and teachers; they all work together as active collaborators of the process. Of course teachers can (and should) stimulate students to develop new ideas, to diverge and to be creative. Therefore, students are assessed throughout the whole process: from ideation to the final deliverable.

The CBL framework is divided into three interconnected phases: *Engage*, *Investigate* and *Act*. Each phase includes a different set of activities:

- ***Engage***:
  Big Idea: a broad concept that can be explored. It has to be a topic that is engaging for students;
  Essential Question: the question related to the *big idea* that students want to explore;
  Challenge: a call to action derived from the essential question. Should be actionable and exciting;
- ***Investigate***:
  Guiding Questions: questions related to the challenge. Includes everything that needs to be learned;
  Guiding Activities and Resources: list of activities and resources that can help students pursue the challenge;
  Analysis: sets the foundation to develop the solution to the challenge;
- ***Act***:
  Solution Development: based on the learnings from the previous steps, the solution is implemented;
  Evaluation: verifies if the solution has addressed the challenge or if it needs refinement;

A key component of the CBL framework is the *reflection.* A significant part of the learning process is built when students take some time to reflect over the activities being done. Not only student have the opportunity to think about the project being undertaken, but they can also reflect about how they are interacting with the team, the teachers, and the environment. This process is usually privately recorded and only teachers may have access to this information.

2.2 Startups and Customer Development

Giardino *et al.* [12] define startup as a small organization that explores new market opportunities by developing a solution to a problem in a volatile market. Blank and Dorf [5] define a startup as a temporary organization that is searching for a sustainable and repeatable business model. Eric Ries [24] defines a startup as "*a human institution designed to create a new product or service under conditions of extreme uncertainty*". Regardless of the formal definition, startups usually are innovative, highly reactive, suffers from lack of resources and time pressure, and most importantly, are highly risky endeavors [11, 28].

In regards to software development processes, startups usually share the same problem: they need to know what to build. Thus, this can become a real challenge, since users/customers might be unknown. Therefore, validating business hypothesis is key to any startup success, and this is done by developing a minimum viable product (MVP) [16]. The MVP, if done correctly, allows startup to collect feedback from users/customers. This feedback is transformed into software requirements, allowing the startup to move forward [11, 12].

However, understanding what "minimal" means can be a challenge [15]. An MVP is not supposed to be a beta version of a product; it should be the smallest effort to learn from users/customers. This is where the customer development process comes into play. Steve Blank [4] proposed and created this process based on the premisse that most startups fail from lack of customers, rather than product development issues. Blank claims that there are tools, methods and frameworks to manage and to control software development. However, there is no process to take care of the customer development.

Figure 1 presents Steve Blank's model [4], which is divided into four steps:

- *Customer Discovery*: defines which problem and customer will be tested and run a set of experiments in order to validate whether the problem is relevant to those customers;
- *Customer Validation*: test and validate the sales process. If there is not enough evidence that people are willing to pay for the solution, the startup should change the strategy. This move is called *pivot*;
- *Customer Creation*: once the business model is validated, it is time to test whether the business can gain traction;
- *Company Building*: formal management is put into place and growth strategies are created. At this point, the organization is no longer a startup.

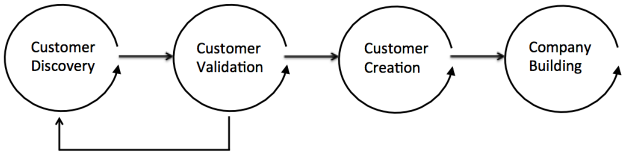

Figure 1: Customer Development Process [4]

From a startup standpoint the first two phases are the most important and the most critical. If the startup is not able to validate its business model before running out of resources, it will go out of business. Hence, it is fundamental to develop an effective customer discovery process plan [2]. Running interviews and testing value propositions on social media and websites are great examples of effective and somehow cheap strategies to undertake.

A big concern startups have is related to software quality and technical debt [11]. In fact, startups face a real dilemma since following a strict software development methodology may lead to waste of time and resources. On the other hand, not following good practices may require a lot of effort in the future in order to keep minimum software quality standards.

Finally, from an education standpoint, technology-related undergraduate and graduate programs are adapting themselves to fit startup content into their curriculum [1, 9]. The challenge lies on providing opportunities for students to develop a real startup that will face customer and market pressure [6]. Working with "toy" projects definitely helps students learning technical content. However, developing a startup is also about critical thinking, problem solving, and adaptability. Hence, these abilities also need to be addressed in an effective learning environment.

3 RELATED WORK

There are a few publications that connect CBL and software development practices [3, 25]. Agile practices are often used in a startup context. In this sense, our work was inspired by Santos *et al.* [25], which presented the first framework that formally combines CBL methodology with agile practices.

Santos *et al.* [25] explored a case study of an iOS development course that used CBL as the methodology for teaching. Even though there was no formal software development process put it place when the first course was offered, it became clear that student would need one. Therefore, a framework was proposed and implemented. In this framework, students do not start implementing their mobile application until they are done with their investigation. If something goes wrong during this step, it might be to late to go back and adapt the solution. Moreover, the model does not make any reference to outside users or customers. Even though this work presents a great methodology for teaching mobile application development, the lack of real users/customers interaction is clearly a gap. Teachers can assess whether students learned how to develop a given mobile application. However, nothing is mentioned regarding the validation of the proposed solution by real users. In order to address this points, the present work introduces an evolution of this model.

4 THE CHALLENGE BASED STARTUP LEARNING FRAMEWORK

As mentioned in Section 3, the combination of CBL and Scrum was very effective in regards to teaching students how to develop mobile applications from a technical point of view. However, the framework fails to address the challenge from the point of view of the people affected by it. Linus Torvalds, the creator of Linux, once said that "*any program is only as good as it is useful*". Thus, it is crucial that students begin the process with this mindset. The goal of a startup, as already mentioned in this work, is to create a sustainable and scalable business. Hence, knowing how to build a mobile application (or any software application for that matter) is not enough; the system has to be useful for a given group of people.

In this sense, the proposed framework is a result of applying CBL, lean startup, customer development and software development strategies, for two years in an undergraduate digital entrepreneurship course. By doing so, learners not only have the opportunity to develop the necessary technical skills, but they can also connect themselves with potential users/customers in order to gain real world experience. It is important to point out that the main goal of this framework is not to teach how to develop a successful startup (even though this would be great for students and the community), but to teach the software startup development process.

Figure 2 presents the Challenge Based Startup Learning framework overview. The process starts in the *engage* phase, where students define their big idea, essential question and the challenge, using design thinking and brainstorming processes. This phase is straightforward and it follows the regular CBL model. It is critical for students to build engagement and excitement. If students are not really connected to the challenge, they should revisit this step; any startup begins from a founder's vision about an idea. If there is not enough excitement about it, students may quickly abandon the project, as it happens in real life [11].

Once this first step is completed, learners can move to the sprints phase. Sprint length as well as the number of sprints should be defined according to the time and the context of the course. In regards to the length, we suggest a minimum of two-week and a maximum of four-week period. As it can be observed in the framework, each sprint combines activities from the *investigation* and *act* phases. At the beginning of this process, naturally, students need to focus more on investigating the proposed challenge as well as the potential stakeholder affected by it. In order to do so, customer development and lean startup strategies can be put in place. As students gather the necessary information and begin to validate their hypothesis, they may plan further sprints focusing more on the execution (*act* phase), *i.e.*, delivering software. However, if students fail in this process and are not able to validate their assumptions, they should *pivot* [5] and remain in the same step until information is validated. It is also possible to move back in the framework in case students understand that more investigation is needed.

In the first sprint, we suggest running interviews in order to gather information regarding the challenge. The goal is to validate whether a group of people carry the same problem. To begin with, potential users/customers need to be defined. Using the concept of personas [8] can help students achieve this task. Once the persona is developed, *guiding questions* and *guiding activities* are designed. The general rules for running interviews are [26]:

- *No leading question*: asking "Do you prefer *A* or *B*?" may lead an interviewee to say *A*, when in fact the person prefers *C*;
- *Ask open-ended questions*: interviewers may find interesting information when asking for stories;
- *No pitching*: it is a moment about learning, not about selling;
- *Ask questions about the past*: people are awful at predicting the future. Past behavior are best predictor of future behavior;

Even though it is not statistically relevant, talking to 20 people should be enough to understand the process and to look for a pattern. If the problem is not validated or no pattern is found, the team should pivot and redesign the interview process. Since we are talking about a learning environment, it is not wrong to move forward in the framework even if assumptions are not validated as long as both students and teachers are aware of this situation.

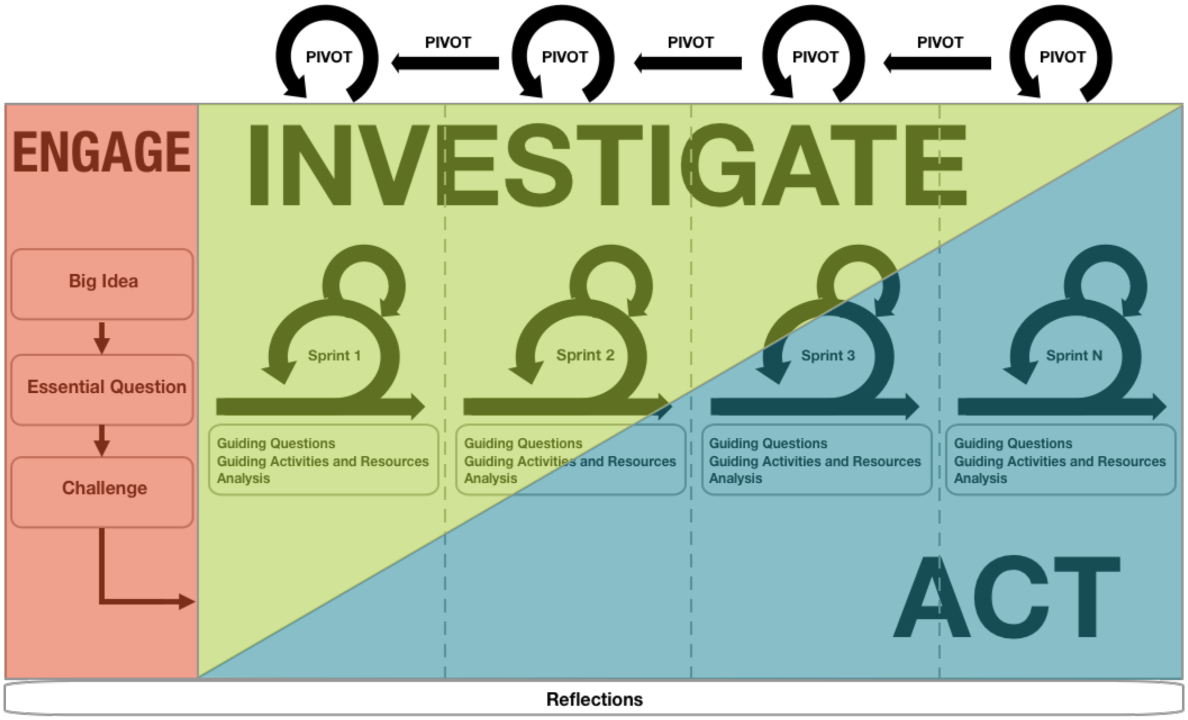

Figure 2: Challenge Based Startup Learning Framework

For the next sprint, we suggest a value proposition testing. Developing a landing page with a call to action process, structuring a social media page, or creating a video are some of the strategies to be put in place. The goal is to learn whether potential users/customers can be attracted, and if they are somehow interested in the value proposition proposed. A big challenge in this step is to find the right channels to promote the action. Once more, if students fail in this experiment, they should pivot and rethink about the strategy.

Creating and maintaining audience is defined as the next step. Designing a blog or a social media channel are two great ways to do it. The goal is to create content that will attract and engage potential users/customers. By succeeding in this step students will have real people to test and to try out the proposed solution.

It is important to point out that all these lean startup and customer development strategies can and should be continuously executed. For instance, if a blog is designed, content should be created on a regular basis in order to keep people interested on the topic.

At this point, students should have enough information to create the first prototype or the first version of the solution. Sprints should be organized in such a way that, at the end of them, there is always something to show to potential users. Even if the solution presents bugs or it is not ideally designed, it is better to collect users' feedback as soon as possible than waiting until the software is "perfect" [24].

As it can be observed in the framework, reflections occur throughout the whole process. At the end of each sprint, it is important to stop for a few moments to reflect over the experience. This material should be used by teachers to improve the process as well as to help students overcome problems and difficulties. This process is done by either recording a video or by writing a small essay.

Lecturers must be aware that students might struggle in finding a topic to work on. If that is the case, it is interesting to have a list of big ideas to present to students. In addition, working with pivots can be risky due to time restrictions of a semester. If students need to pivot in one of the proposed steps, we suggest two alternatives: (i) give students more time to work on that step and reduce the time at the end of the semester for the solution development; (ii) explain to students that they should have pivoted if this was a real project. However, they need to move on to the next phase so they can experience the whole startup creation process.

Regarding assessment, the focus is always on the process rather than the final result. In other words, even if students fail in collecting enough responses from interviews, for instance, they should be graded based on how well the interview was designed and conducted, and not by the number of responses.

5 FRAMEWORK APPLICATION AND PRELIMINARY RESULTS

The proposed framework was applied in an undergraduate digital entrepreneurship course that lasted one semester. Table 1 presents the schedule overview of the course. At the end of each activity, students also deliver a reflection document. The semester started by presenting the basic concepts (Challenge Based Learning, Lean Startup, Customer Development and Scrum). Right after this moment, the class was divided into five teams of three to four students and groups began to work on their *Big Ideas*, *Essential Question*, and *Challenge*. The result of this process is presented in Table 2.

Table 1: Schedule overview.

Week	Activity	Deliverable
1	Basic concepts	Exercises
2	Engage phase	Big Idea, Essential Question, and Challenge
3–4	Sprints	Interview results
5–6	Sprints	Value proposition testing results
7–8	Sprints	Content creation
9–12	Sprints	Low-fidelity prototype
13–16	Sprints	High-fidelity prototype

Once the challenge was defined, teams moved to the sprints phase. Sprint length was set to two weeks so we could have seven sprints in total. In the first sprint, teams designed their strategy to gather information through interviews and observations. Some groups developed online surveys, while others decided to personally talk to potential users. A key learning point was that even though online surveys can bring more data, personal interviews bring more knowledge, since students can interact with people and go to different directions depending on the answers they get. Even though all teams concluded that they needed more data to support their findings, we decided to move to the next experiment so they could learn and experience other techniques.

Table 2: Engage phase.

Big Idea	Essencial Question	Challenge
Tourism	What people look for when visiting another country?	Deliver a great experience for people visiting St. Barth.
Charity	What makes people engage in charity events?	Make donation easier for everyone.
Finance	How does the use of cash impact the life of students?	Make payments easier.
International Culture	How does people interact when visiting another country?	Make connections that matter.
Entertainment	What people look for when going out?	Deliver the best venue option according to your taste.

In the following sprint, teams were required to design a landing page presenting their value proposition with a call to action process (all teams created a form to collect information, such as name and email). Then, teams needed to announce and promote their page in order to test acceptance. Results are presented in Table 3. The data in the table shows the number of page views each project obtained and the correspondent conversion rate (percentage of people that filled out and submitted information through the form). All teams were very disappointed with the results. However, the biggest learning happened here; students realized it is not easy to get someone's attention.

The next sprint was focused on content creation. Teams had to decide on developing either a blog or a Facebook page in order to develop a community around their idea. Although this activity began in the third sprint, teams were asked to generate content until the end of the semester. As mentioned in Section 4, the goal of this process is to engage people into following the project.

The goal of the last four sprints was to develop a working version of the proposed solution in order to collect feedback from users. The backlog for each project was organized is such a way that by the end of the fifth sprint an MVP had to be available in production (a low-fidelity prototype). Three teams (the tourism, the charity and entertainment ones) were able to meet this deadline. The other two groups were only able to deliver their first version of the project on the sixth sprint. Bad backlog management was the cause of such delay. It was no coincidence that the groups that manage to deliver earlier were able to get real users feedback and improved their solutions, while the other two groups did not have time to do so.

Table 3: Call to Action Results.

Big Idea	Page Views	Conversion
Tourism	422	2.4%
Charity	276	0.36%
Finance	312	2.6%
International Culture	188	1.6%
Entertainment	205	4.4%

5.1 Survey

In order to evaluate the framework, an individual survey was conducted at the end of the *Engage* phase and after sprints one, two, three, five and seven. The main goal was to verify whether students understood the process of creating a real startup. The sample population was the total number of students in the class (18 students), and questions were designed to cover students' perceptions of the whole process as well as of each deliverable (interview process, landing page, content creation, and the solution). Questions were asked on a six–point Likert scale (totally disagree, disagree, somewhat disagree, somewhat agree, agree, and totally agree). The choice for an even-numbered scale was to avoid neutral positions. In addition, we also analized the data from students' reflections. At the end of each sprint students were asked to document their perceptions and thoughts.

In regards to the *Engage* phase, 83% of the students totally agreed and 17% agreed that the CBL process helped them finding an engaging challenge to work on. From a startup point of view, this is very important since founders need to share the same vision and need to be equally engaged in the project.

After the first sprint, students were asked a question about the interview process and another one about their motivation so far. In this case, 94% of the students totally agreed that the interview process was very helpful in understanding people's perception over a given problem. Regarding their motivation, 55% agreed and 45% somewhat agreed that they were satisfied and motivated with the results so far. Further analysis indicated that some students were frustrated about the difficulties in getting responses. This point was later explored in the classroom to show students how hard it is to grab people's attention; and running a startup is all about that.

The landing page development activity survey presented interesting results. All students totally agreed that performing a value proposition testing was a great way to play with conversions. Even though conversion rates were below their expectations, motivation and engagement went up at the end of the third sprint.

Regarding content creation, one student somewhat disagreed, ten students somewhat agreed, and seven students agreed that this activity was helpful for the startup process development. At this point, students could not see the value of creating a community (that would be further testing and evaluating their systems).

At the end of the fifth sprint, as already mentioned, three teams managed to have a working version of their proposed solution, while two teams could not meet the deadline. Interestingly, 95% of students totally agreed and 5% agreed that having gone through a few experiments before building the software gave them good insights and information to better define the product backlog as well as to correctly prioritize tasks. All students were highly engaged and motivated by the process.

Finally, the overall perception of the Challenge Based Startup Learning framework was that it gave students a great experience on what it takes to run a real startup. All students totally agreed on that statement. Besides, some students mentioned that it was very engaging to work on projects that can really make a difference in people's lives.

6 FINAL REMARKS

We presented in this work the Challenge Based Startup Learning, a framework that combines Challenge Based Learning, lean startup, customer development and software development techniques. The framework is a result of using all these methodologies for two years in an undergraduate entrepreneurship course. Our preliminary results indicate that the framework can help students experience the process of creating a real startup.

It is highly important to point out that the intention of this work is to introduce the first insights and thoughts regarding our proposition. Further investigation and research still needs to be done in order to test other variables. We are aware that there is not enough data to support the validity of the framework yet.

Developing a startup requires not only technical skills, but also a lot of soft skills. It is very important for students to understand that they need to explore and learn about customers and their problems. There are several tools and strategies available to validate assumptions before putting a lot of effort on coding. By running the right experiments, students can develop a better vision regarding requirements and backlog prioritization.

As future work, we intend to collect more data by applying the proposed framework on different educational settings and contexts. Our main goal is to find an effective and engaging way for students to learn what it takes to run a real startup. Moreover, we expect to inspire and encourage students to create solutions that really impact our society.

ACKNOWLEDGMENTS

This work is partially funded by FAPERGS (17/2551-0001/205-4).

REFERENCES

[1] I. Aaen and J. Rose. 2011. A Software Entrepreneurship Course: Between Two Paradigms. In *Proceedings of the 15th Annual Interdisciplinary Entrepreneurship Conference*. Research School Business and Economics, St. Gallen and Zurich, Switzerland.

[2] T. Batova, D. Clark, and D. Card. 2016. Challenges of Lean Customer Discovery as Invention. In *2016 IEEE International Professional Communication Conference (IPCC)*. 1–5.

[3] F. V. Binder, M. Nichols, S. Reinehr, and A. Malucelli. 2017. Challenge Based Learning Applied to Mobile Software Development Teaching. In *2017 IEEE Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, 57–64.

[4] S. Blank. 2013. *The Four Steps to the Epiphany: Successful Strategies for Products That Win.* K&S Ranch, Incorporated.

[5] S. Blank and B. Dorf. 2012. *The Startup Owner's Manual: The Step-by-step Guide for Building a Great Company.* K&S Ranch, Incorporated.

[6] K. Buffardi, C. Robb, and D. Rahn. 2017. Tech Startups: Realistic Software Engineering Projects With Interdisciplinary Collaboration. *Journal of Computing Sciences in Colleges* 32, 4 (2017), 93–98.

[7] G. Coleman. 2005. An Empirical Study of Software Process in Practice. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS)*. IEEE Computer Society, Big Island, HI, USA, 315c 1–6.

[8] A. Cooper. 1999. *The Inmates Are Running the Asylum.* Macmillan Publishing Co., Inc., Indianapolis, IN, USA.

[9] K. Daimi and N. Rayess. 2008. The Role of Software Entrepreneurship in Computer Science Curriculum. In *Proceedings of the 2008 International Conference on Frontiers in Education: Computer Science & Computer Engineering (FECS 2008)*. IEEE Computer Society, Las Vegas, NV, USA, 332–338.

[10] C. Giardino, N. Paternoster, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson. 2016. Software Development in Startup Companies: The Greenfield Startup Model. *IEEE Transactions on Software Engineering* 42, 6 (2016), 585–604.

[11] C. Giardino, M. Unterkalmsteiner, N. Paternoster, T. Gorschek, and P. Abrahamsson. 2014. What Do We Know about Software Development in Startups? *IEEE Software* 31, 5 (2014), 28–32.

[12] C. Giardino, X. Wang, and P. Abrahamsson. 2014. *Why Early-Stage Software Startups Fail: A Behavioral Framework.* Springer International Publishing, Paphos, Cyprus, 27–41.

[13] L. Johnson and S. Adams. 2011. *Challenge Based Learning: The Report from the Implementation Project.* Technical Report. The New Media Consortium, Austin, TX, USA.

[14] M. Kajko-Mattsson and N. Nikitina. 2008. From Knowing Nothing to Knowing a Little: Experiences Gained from Process Improvement in a Start-Up Company. In *International Conference on Computer Science and Software Engineering (CSSE 2008)*. Wuhan, China, 617–621.

[15] V. Lenarduzzi and D. Taibi. 2016. 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)*. 112–119.

[16] S.-C. Li. 2007. The Role of Value Proposition and Value Co-Production in New Internet Startups: How New Venture e-Businesses Achieve Competitive Advantage. In *Portland International Center for Management of Engineering and Technology (PICMET)*. Portland, OR, USA, 1126–1132.

[17] C. Marin, J. Hargis, and C. Cavanaugh. 2013. iPad Learning Ecosystem: Developing Challenge-Based Learning Using Design Thinking. *Turkish Online Journal of Distance Education* 14, 2 (2013), 22–34.

[18] A. Nguyen-Duc, P. Seppänen, and P. Abrahamsson. 2015. Hunter-gatherer Cycle: A Conceptual Model of the Evolution of Software Startups. In *Proceedings of the 2015 International Conference on Software and System Process (ICSSP 2015)*. Tallinn, Estonia, 199–203.

[19] M. Nichols and K. Cator. 2008. *Challenge Based Learning White Paper.* Technical Report. Apple Inc., Cupertino, CA, USA.

[20] M. Nichols, K. Cator, and M. Torres. 2016. *Challenge Based Learning Guide.* Nichols, M. and Cator, K. and Torres, M., Redwood City, CA, USA.

[21] U.S. Department of Commerce. 2013. *The Innovative and Entrepreneurial University: Higher Education, Innovation and Entrepreneurship in Focus.* Technical Report.

[22] N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson. 2014. Software Development in Startup Companies: A Systematic Mapping Study. *Information and Software Technology* 56, 10 (2014), 1200–1218.

[23] J. Porter, J. Morgan, R. Lester, A. Steele, J. Vanegas, and R. Hill. 2015. A course in innovative product design: A collaboration between architecture, business, and engineering. In *Proceedings of 2015 IEEE Frontiers in Education Conference (FIE)*. IEEE Computer Society, El Paso, TX, USA, 1–5.

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

[25] A.R. Santos, A. Sales, P. Fernandes, and M. Nichols. 2015. Combining Challenge-Based Learning and Scrum Framework for Mobile Application Development. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '15)*. ACM, Vilnius, Lithuania, 189–194.

[26] J. Scheinrock and M. Richter-Sand. 2013. *The Agile Startup: Quick and Dirty Lessons Every Entrepreneur Should Know.* John Wiley & Sons.

[27] K. Schwaber. 1995. SCRUM Development Process. In *Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA)*. ACM, Austin, TX, USA, 117–134.

[28] S. M. Sutton. 2000. The role of process in software start-up. *IEEE Software* 17, 4 (2000), 33–39.