A Collaborative Approach to Teaching Software Startups: Findings From a Study Using Challenge Based Learning

Rafael Chanin, Afonso Sales, Alan Santos, Leandro Pompermaier, Rafael Prikladnicki
Pontifical Catholic University of Rio Grande do Sul, School of Technology
Av. Ipiranga 6681, Porto Alegre, Brazil
{rafael.chanin,afonso.sales,alan.santos,leandro.pompermaier,rafaelp}@pucrs.br

ABSTRACT

Education on software startups is taken its first steps. A multidisciplinary and collaborative approach seems to be key to deliver a high quality software startup development program. In this sense, we conducted a study describing the path of 191 students through a startup mobile application development program. Our preliminary results indicate that using Challenge Based Learning methodology combined with agile practices can strengthen students' collaboration and engagement into the process.

CCS CONCEPTS

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

KEYWORDS

Challenge Based Learning; Software Startup; Mobile Application; Software Development; Collaboration; Engagement.

ACM Reference Format:
Rafael Chanin, Afonso Sales, Alan Santos, Leandro Pompermaier, Rafael Prikladnicki. 2018. A Collaborative Approach to Teaching Software Startups: Findings From a Study Using Challenge Based Learning. In *CHASE'18: CHASE'18:IEEE/ACM 11th International Workshop on Cooperative and Human Aspects of Software , May 27, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3195836.3195844

1 INTRODUCTION

The popularization of the Internet and evolution of the technology in the past decades have changed the way society interacts [7]. Nowadays, through the use of a smartphone people can communicate with each other at any point in the planet, pay bills, listen to music and even watch their favorite TV show. Companies such as Facebook, Google, Whatsapp and Dropbox, which have over one billion users around the globe, could only exist thanks to all this technology growth.

These new technology ventures, that work in a high uncertainty environment and aim to find a sustainable and repeatable business model, are call *startups* [1]. The majority of the startups follow the *lean startup* methodology [10]. This approach combines short software development cycles with constant interaction with potential customers. This process intends to minimize risks, since it focuses on constant learning.

Startups are collaborative endeavors. A multidisciplinary team is a must to achieve success. However, most startups fail in their first years of life [4]. Lack of resources, uncertainty and competition definitely contribute to it. Nonetheless, the lack of support in software engineering processes is pointed as one of the main reasons for startup failures [2, 4, 6].

From an education perspective it is crucial to implement a collaborative and engaging approach in order to help students understand what is takes to build a real startup. Some methodologies, such as Challenge Based Learning (CBL) [9], can fill this gap. In addition, CBL fits into educational environments [11], contributing on the startup formation.

In this paper, we present findings from a startup mobile application development program that uses CBL to teach software development related concepts as well as business and entrepreneurship. The main goal is to evaluate whether CBL can foster collaborative learning as well as engage students into the startup software development process.

The remainder of this paper is organized as follows. Section 2 presents the Challenge Based Learning framework. Section 3 describes the methodology used in this research. In Section 4 we present our preliminary results and, finally, we draw our conclusion and future works in Section 5.

2 CHALLENGE BASED LEARNING

Experiential learning is the source of a variety of learning frameworks that are used all over the world. Problem-Based Learning, Project-Based Learning, Task-Based Learning and Challenge Based Learning are just a few examples of these frameworks.

Challenge Based Learning (CBL) [9] is a learning framework based on solving real world challenges. CBL was developed by educators working with *Apple Inc.* [8] and has been implemented both in educational and corporate environments. From the education perspective, students acquire knowledge by working on open-ended problems in collaborative teams.

In the CBL learning process, students, professor and other stakeholders work together as active collaborators, and creative and divergent thinking is stimulated. In addition, the focus is not only on the final deliverable, but also on the whole process; students must reflect from time to time on their learning evolution.

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

CHASE'18, May 27, 2018, Gothenburg, Sweden
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5725-8/18/05…$15.00
https://doi.org/10.1145/3195836.3195844

- ***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. It 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.

Table 1 illustrates some examples of the *Engage* phase. As it can be noticed, *Essential Questions* are always designed as questions, whereas *Challenges* are statements.

Table 1: Engage phase.

Big Idea	Essencial Question	Challenge
Tourism	What people look for when visiting another country?	Deliver a great experience for people visiting Canada.
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.

Johnson and Adams [5] have demonstrated that the use of active learning methodologies improves students' learning when compared to traditional methods. Moreover, the engagement and the soft skills acquired during the process is also viewed as a big asset for students and the stakeholders involved.

3 METHODOLOGY

The research methodology for this study was based on a process proposed by Eisenhardt [3]. The following sections describe and explore each of the steps undertaken.

3.1 Getting Started

The goal of this work is to identify whether CBL can strengthen students' collaboration and engagement when learning about software engineering processes in the context of software startup development. In this sense, the research question proposed is *Can Challenge Based Learning help students improve their software engineering skills when developing software startups?*

In order to answer this question, we explored a mobile application training program as our case study. The case is detailed in the next section.

3.2 Case Study

The context we have studied is a two-year mobile application training program focused on iOS technology. Curriculum includes the following topics: Object-Oriented Programming, User Interface (UI) Components, Model View Controller, Data sources, Navigation, Animations and Frameworks, the Scrum framework [12], and Startup related content. After taking theoretical lessons, all students work on developing real world mobile applications using agile practices to support it. In this sense, the program was organized as follows:

- First year: focused on technical aspects related to mobile application development. During this period students learn Object-Oriented Programming, Model View Controller, Datasources, Navigation, UI components, Animations and Frameworks, and Scrum.
- Second year: focused on design and entrepreneurship. Even though students continue to learn technical content, the goal is to improve design techniques as well as to learn how to build an application from a business perspective.

The whole program is CBL-centered. Students learn the concepts through active learning methods and constant feedback and reflections. The faculty and students work together throughout the whole process, from ideation to the solution implementation. Each step may require a different expertise. This is why specialized instructors with different backgrounds are needed. Their knowledge includes application and software development, design, entrepreneurship, business and project management. By the end of 2016, the program have graduated 191 students. All of them went through our survey process.

3.3 Data Collection

During this two-year period, each student individually documented his/her learnings based on a CBL concept called *reflection* [9]. Once a month, students either recorded a video or wrote a document reflecting on the content they have learned as well as on the process they were going through. This process not only helped them reflect on their evolution in the program, but also gave instructors a great amount of inputs to adjust the program for the following month. Every single reflection had a guiding question that should be addressed.

In addition to the reflections, we collected another set of information from students in the beginning and at the end of the program (anonymously). Questions were focused on four main areas:

- General and demographic information;
- Technical (programming) knowledge;
- Soft skills and project management experience;
- Challenge Based Learning and entrepreneurship.

General and demographic information were collected through a structured questionnaire, while the remaining information were asked using a scale from 0 to 10.

3.4 Analysing Data

Once all data was collected, we searched for patterns and interesting insights. This information was categorized in order to make it easier to answer our research question.

It is worth mentioning that our evaluation is based on data collected from a questionnaire and from students' reflections. Researchers might have interpreted the collected data according to their expectations. In order to mitigate this threat, we presented the collected information to other stakeholders (such as the program instructors) in order to verify the results.

This was done by using a triangulation approach: we included researchers with different roles into the data collection, and we cross-checked the data with the reflections and with instructors' perceptions. However, we can not guarantee that results are more reliable just because we applied this approach. The goal was to minimize any potencial bias from the authors.

4 RESULTS

We have found indications that this program exceeded students expectations (see Table 2). Having an average overall satisfaction of 9.3 out of 10, with a standard deviation of 1.08, indicates that students' perception were similar. In addition, students have published 210 applications in the online stores. This number is twice as big as the number of applications actually required to be developed by the program activities. This indicates how proactive and engaged students became throughout the program.

Table 2: Program demographic information

Average age when joined the program	22
Gender	
Male	90%
Female	10%
Expectation before joining the program (1-10)	7.8
Overall satisfaction after the program (1-10)	9.3
Total graduated students	191
Total apps published	210

Another interesting fact worth mentioning is that students do not come only from computer science related courses (see Figure 1). We believe that this broad range of students' backgrounds and interests have also contributed to the success of the program. Multidisciplinary teams are crucial to reduce failure rates in software startups [4].

In regards to technical software development learnings, students presented a significant evolution (Table 3). All students joined the program knowing basic programming skills, but none of them knew how to develop mobile applications. By analyzing the most mentioned technical learning topics, we can notice that students valued software quality and testing. Taking into account that participants were somehow young (22 years old on average), this feedback shows how mature students became in regards to the software development process.

Students also presented improvement in their soft skills and project management knowledge (Table 4). This fact can also be confirmed by reading through students' reflections. One student said: *"I know I have the classic Computer Science stereotype (focused on technical skills). However, the program helped me change my perception over soft skills. Now I can see the importance of understanding people's behavior."*.

Table 3: Technical Knowledge

Programming knowledge before joining the program (1-10)	6.0
Programming knowledge after the program (1-10)	8.4
Most mentioned technical learning topics (# of mentions)	
User Interface	77
Model View Controller	63
Navigation	63
Data sources	47
Software Quality	34
Software Testing	15

Throughout the program, students were always organized in teams in order to work on their challenges. For every challenge, teams needed to present their results to the whole class as well as to instructors. This setup helped students understand not only the importance of defining clear roles and responsibilities but also to solve problems in a collaborative manner as soon as their arise. In addition, by having to present their projects to one another on a regular basis, students felt peer pressure; teams were constantly trying to raise the bar in order not to fall behind one another.

Table 4: Soft Skills and Project Management

Agile knowledge before joining the program (1-10)	4.2
Agile knowledge after the program (1-10)	7.6
Leadership experience before joining the program (1-10)	4.6
Leadership experience after the program (1-10)	7.5
Teamwork experience before joining the program (1-10)	6.0
Teamwork experience after the program (1-10)	8.5
Conflict resolution experience before joining the program (1-10)	5.2
Conflict resolution experience after the program (1-10)	7.6
Presentation skills before joining the program (1-10)	5.2
Presentation skills after the program (1-10)	8.7
Creative thinking before joining the program (1-10)	5.9
Creative thinking after the program (1-10)	8.1

Table 5 summarizes our findings regarding the use of Challenge Based Learning throughout the program.

Table 5: CBL and Entrepreneurship

Impact of CBL on the learning process (1-10)	Impact of CBL on the student's engagement (1-10)	How prepare I fell to work for a startup (1-10)	How prepare I fell to create my startup (1-10)
7.9	8.1	9.0	9.1

By analyzing the information from the questionnaire and the reflections, we understand that CBL was key for students to reach the maturity level they achieved after the end of the second year.

Since they were all working with real life challenges, a strong feeling of identification and belonging was developed. This led to a more effective learning process, since students really wanted to deliver the best possible application to their potencial users. For instance, one group developed an application to improve the life of children with Rett Syndrome. Once they began interacting with doctors, parents, and also the children, the level of commitment and engagement went up significantly. The team's interest in learning about software quality, design and user experience, to name a few topics, became higher when outside stakeholder started to get involved in the project.

Students' reflection also gave us interesting insights regarding collaboration. With no exception, all students emphasized how they ended up understanding the importance of working effectively within their groups as well as among other groups. The data in Table 4 corroborates with these findings. In fact, students embraced collaboration and open learning once they recognized and understood the value of it. This is interesting since most students came from the computer science field study (Figure 1).

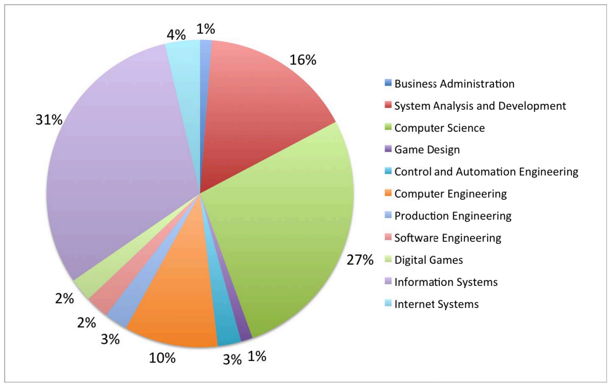

Figure 1: Students' undergraduate courses

From an educational standpoint, we were able to find out interesting information that can help instructors and professors when teaching software startups. First, learning about software startup development can be exciting as well as frustrating. As already mentioned in Section 1, most startups fail. Even though failure is part of the process, instructors must find ways to keep students engaged. One way to do so is to connect them with industry experts, startup founders or any other stakeholder that can contribute to the projects.

Second, even though instructors should not interfere in teams formation, it is worth highlighting the importance of working in a multidisciplinary team. In addition, the best teams in term of collaboration, coordination and results had four or five members. Working with more than five members required a lot of synchronicity, whereas having less than four people usually culminated in a poor team composition.

Finally, having teams presenting their progress once a month proved to be effective. It was a great opportunity to receive and to give feedback to each other. Hence, this approach definitely helped students developing their presentation and communication skills.

5 CONCLUSION AND FUTURE WORK

Teaching software startups is challenging. Students definitely need a practical approach that can help them understand the process of creating and developing a real startup. In this sense, the combination of Challenge Based Learning and software development processes was proposed in order to achieve this goal.

Challenge Based Learning seems to fit well into a software startup educational context, since its core is based on real-world problem solving. Students need to work in collaborative teams in order to investigate the proposed challenge and to find a solution.

In this work, we have analyzed the path of 191 students that went through a startup mobile application development program that uses Challenge Based Learning as the teaching methodology. Our preliminary results indicate that Challenge Based Learning can help students embrace collaboration and open learning towards the software startup learning process.

As future work, we intend to identify other factors that can foster as well as jeopardise the learning process by using CBL to teach software development related topics. Nevertheless, we believe that using an active learning methodology that combines collaborative and engaging approaches makes sense when teaching problem-solving related topics, such as software startups.

ACKNOWLEDGMENTS

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

REFERENCES

[1] 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.
[2] G. Coleman. 2005. An Empirical Study of Software Process in Practice. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS).* Big Island, HI, USA, 315c–315c.
[3] Kathleen M Eisenhardt. 1989. Building theories from case study research. *Academy of management review* 14, 4 (1989), 532–550.
[4] 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 (June 2016), 585–604.
[5] L. Johnson and S Adams. 2011. *Challenge Based Learning: The Report from the Implementation Project.* Technical Report. The New Media Consortium, Austin, TX. Retrieved on April 23, 2014, from http://www.nmc.org/news/challenge-based-learning-report-implementation-project.
[6] 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)*, Vol. 2. Wuhan, China, 617–621.
[7] 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.
[8] M. Nichols and K. Cator. 2008. *Challenge Based Learning White Paper.* Technical Report. Apple Inc., Cupertino, CA, USA. http://cbl.digitalpromise.org/wp-content/uploads/sites/7/2016/12/CBL_Paper_2008.pdf
[9] M. Nichols, K. Cator, and M. Torres. 2016. *Challenge Based Learning Guide.* Digital Promise, Redwood City, CA, USA.
[10] E. Ries. 2011. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses.* Crown Business.
[11] 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).* Vilnius, Lithuania, 189–194.
[12] Ken Schwaber. 1995. SCRUM Development Process. In *Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA).* Austin, Texas, USA, 117–134.