# Innovation and Active Learning for Training Mobile App Developers

Fabio Vinicius Binder
PPGIa
PUCPR
Curitiba, Brazil
fabio.binder@pucpr.br

Regina Albuquerque
PPGIa
PUCPR
Curitiba, Brazil
regina.fabia@pucpr.br

Sheila Reinehr
PPGIa
PUCPR
Curitiba, Brazil
sheila.reinehr@pucpr.br

Andreia Malucelli
PPGIa
PUCPR
Curitiba, Brazil
malu@pucpr.br

## ABSTRACT

The mobile market is growing every year, creating a demand for mission-critical and innovative enterprises, and personal applications. This demand implies the need for skills training necessary for students to enter the professional development market. The Innovation, Development and Teaching in Complex Software Environments approach (IDEAS) was developed for the training of mobile developers to meet this need, through an innovative process for student selection, training through the active methodology Challenge Based Learning, software production, and startups creation. The approach was created through action-research applied during two cycles of a real project in a university environment, where 398 students applied for the first cycle, of which 110 were selected, and 900 applied for the second cycle, of which 100 were selected. We undertook evaluations with students and teachers through periodic reflections, questionnaires, and interviews at the end of each cycle. Some of the positive results included the prominence of 125 professional-looking apps that were finalized and published, report of dropout reduction and disapproval in the programming courses and experience in entrepreneurship and pre-acceleration startup program.

## KEYWORDS

Active Learning, Challenge Based Learning, Mobile Development, Programming Learning, Programming Teaching

## 1 Introduction

The mobile phone market (smartphones) is growing at an accelerated rate, creating a strong demand for personal and business applications (apps) [1]. However, the development of apps for mobile devices has brought challenges to the programming area, which had to be modified and updated. Moreover, projects increasingly involve innovation and have become more challenging for everyone involved in the development cycle. Beyond the completely different interface and usability features, apps have shorter life cycles [35] when compared to the software designed for a desktop/laptop or a website, and use a different form of distribution, which is pre-installation on the device or download from an application (app) store [2]. Apps are becoming integral in companies' mission-critical software portfolio [1]. This demand implies the increasing need for solid training of mobile app developers, which offers basic programming, apps' development, and software development for publishing [3].

The programming courses have significant challenges, such as high dropout. One of the main challenges is the provision of skills training required by the students to enter the professional apps development market, which not only includes programming, but also software development processes, project management, and testing skills [4]. Hawthorne et al. [5] emphasize that a set of skills and competences are needed far beyond software development, and cite the examples of product and project management, and communication skills with people from different cultures to develop distributed systems.

The needs and challenges were evidenced when a technological innovation project was set up within a university environment to meet the training demands of mobile app developers. This project involved several universities in Brazil and two other countries, and was supported by a partner technology company. Moreover, the research environment was considered complex due to the following reasons: i) use of a teaching method unknown by teachers and without any literature in relation to its use in teaching programming; ii) heterogeneous characteristics of the students, who came from technical, undergraduate, specialized, master, and doctoral levels; iii) 5 full-time teachers, 5 part-time teachers, and 30 external teachers proposing a project; iv) management of four support teams, comprised of a responsible professor and 5 to 10 scholarship students in the areas of usability, interface, illustration, three dimensional (3D) modeling, animation, sound, soundtracks, innovation, and entrepreneurship; and v) mandatory


ACM ISBN 978-1-4503-7124-7/20/05…$15.00
https://doi.org/10.1145/3377814.3381713

development of innovative and professional-looking apps which were ready for publication.

Thus, this research proposes an approach for a complex environment of mobile app developers' training involving active methodologies and innovation, generating practical knowledge to solve a specific problem. The motivation to use active methodologies is derived from observation and experimentation of student behavior in traditional classes. There is a relationship between the type of stimulus received and the involvement of the students during class. This is evidenced through classroom experience, where students often prefer to engage in practical activities when compared to listening [6]. Active methodologies, primarily the Problem Based Learning (PBL) [7] and Flipped Classroom (FC) [6], or both simultaneously [8], have been used for several years with mixed results in programming courses. An action-research was developed to attain the goal of this study, through the creation of an approach for training developers using the active methodology of Challenge Based Learning (CBL) [9].

This study is organized as follows: the section 2 describes the theoretical framework; section 3 describes the research method; Sections 4 and 5 presents and discusses the results of cycle 2 of the action-research; and Section 6 presents the conclusions.

## 2 Theoretical Framework

### 2.1 Active Learning Methodologies

Active methodologies are a form of teaching with an emphasis on constructivist qualities [10], which has been globally prominent and are recurrently included in long-term educational policies in various countries [11].

Although they do not solve all the problems of education, studies on active methodologies present better results when compared to the studies on the model of expository classes [12] [13], particularly among educators who seek alternative means of teaching in order to increase students' interest [14]. Freeman et al., [15] reported student performance from a meta-analysis of 225 studies on the test scores and failure rates for undergraduate courses of science, technology, engineering, and mathematics. The results of this study indicated a 6% improvement in academic test scores in classes that used active methodologies, while the students in classes taught using the traditional methodology had 50% more failures when compared to the students in classes that used active methodologies.

Active methodologies require the students to both, accomplish meaningful activities and reflect on what has been done [16], thus avoiding the passivity of receiving information, which is very common in traditional classes. The main objective of active learning is to engage students in activities that enable critical thinking of the content being studied. Through active behavior, students are prepared to deal with professional issues and challenges, which require assessment and problem-solving skills.

There are several active methodologies with different approaches, though with common characteristics, such as the active impact of the learner and their involvement in the teaching-learning process [10]. Some examples of these methodologies are PBL [17], which uses real-life problems to engage students in a process of knowledge and solution seeking, where teachers act as facilitators of the process without offering ready answers; Project Based Learning - PjBL [18], which begins with a product that requires certain knowledge for its understanding, thus providing a practical context for engaging students in the study; FC [19], which is the opposite of a traditional classroom, and enables students to study materials outside the classroom while class hours are used to assimilate knowledge through problem-solving and discussion; and CBL [9], which is a motivational and collaborative approach that encourages the use of common technologies for knowledge acquisition and real-world problem solving.
It is believed that active learning methodologies can be used to teach any type of content due to the degree of motivation they provide [15]. Considering its eminently practical nature, teaching programming and software development could benefit from the systematization and organization that the methodology offers. The next section provides details of the active methodology of CBL, which is used in this research.

### 2.2 CBL

The CBL method focuses on students, the teaching-learning community, and common skills of the 21st century [9]. The method has a set of phases with the main purpose to aid solving a critical global problem through local actions. Following are the phases in the CBL method [20]:

- Engage:
    1) Big Idea;
    2) Essential Question;
    3) Challenge;
- Investigate:
    4) Guiding Questions;
    5) Guiding Activities and Resources;
    6) Analysis;
- Act:
    7) Solution;
    8) Implementation; and
    9) Evaluation.

The "Big Idea" is the definition of the central theme of the project, and the "Essential Question" is elaborated from a set of questions that reflect the current global problems related to the theme. Subsequent to reflection and dialogue, team members are expected to reduce these issues to only one major issue, which is intended to be addressed as the "Challenge." "Guiding Questions" and "Guiding Activities and Resources" are intended to better clarify the problem and the skills needed to solve the problem. The "Solution" is the conceptualization of the solution and "Implementation" is the solution construction. Finally, "Evaluation" is the assesment which could be performed by adopting the following optional methods [9]: informative evaluation of the learning process and products generated; documentation and publication of student reflections; or dialogue of students based on their learning.

Considering that one of the objectives of this research is to generate practical knowledge to solve a specific problem, these aspects make the CBL an appropriate method to adopt. Although several studies have used active methodologies in teaching programming, none of them have used it in a complex environment such as in this research.

## 2.3 Active Learning Methodologies in Teaching Programming

Previous studies that used active teaching methodologies in programming courses, primarily PBL methodologies [22], FC [6], and both simultaneously [23], presented mixed results. Considering the positive side, several factors were identified in these studies, such as: students' motivation and consequent engagement during the class; the students following their own pace; collaboration among students naturally occurs due to the dynamic nature of practical activities; proximity between the teacher and students; and a well spent time in class. The negative aspects are related to the students' behavior and effort required to make the classes productive. Students are expected to prepare for classes, put in more effort considering that there is considerably more work to be completed, and should be self-motivated [24]. Teachers also had more work when compared to the same classes being taught using traditional methods [10].

Santos et al. [3] reported a teaching experience using the CBL method combined with Scrum. A six-month iOS app development course was conducted for 94 undergraduate students from different academic backgrounds. The course curriculum consisted of object oriented programming and basic iOS structures. The format of the course involved learning of programming concepts applied to mobile apps and followed the process of CBL, based on the tasks directed to challenge the students. The main finding was that combining the CBL method with Scrum was an effective method to teach the development of apps, because they are complementary, where Scrum helps running challenges and CBL helps in improving the quality of the software developed.

Another experience in this context was the course "Mobile Phones Application Development and Entrepreneurship"[25], which proposed "Just-in-Time Context," a methodology of self-paced teaching. This study aimed to understand the challenges encountered by the students while developing mobile apps and to explore the students' overall Scrum experience by assessing their learning curve, perception, and challenges. The result was considered positive by all those involved in the process - students and instructors who were provided two roles, Scrum Master and the client. Three complete and functional versions of the proposed system (restaurant orders control) were produced, and the group considered the use of Scrum and the students' commitment as the main reasons for the success. Thus, the study demonstrated that the Scrum method could be used in a classroom experience with novice students, with the aim of increasing the chances of producing quality software within the stipulated timeframe. A practical contribution of the research was a guide to introduce Scrum into courses teaching programming.

The research developed and presented in this study has several different characteristics when compared to previous studies, and in addition to requiring the use of active methodology for teaching-learning, it requires an approach that encompasses the student selection process for deploying apps to an app store. This research has challenging characteristics, such as longer course duration, projects chosen by the students themselves, apps published and available in the market, basic teaching of programming, resources produced by the support teams, in-depth notions of entrepreneurship, support for specialized professionals, and support for the creation of startups.

## 3 Methodology

This study aimed at proposing an approach for the complex environment of mobile app developers' training involving active learning and innovation. Thus, considering the collaborative nature of this work and the researcher acting in a concurrent manner for its execution, the method chosen was action-research.

In relation to the work process, this research used the iterative and incremental approach proposed by Coughlan and Coughlan [26], which is composed of three stages, as illustrated in Figure 1.

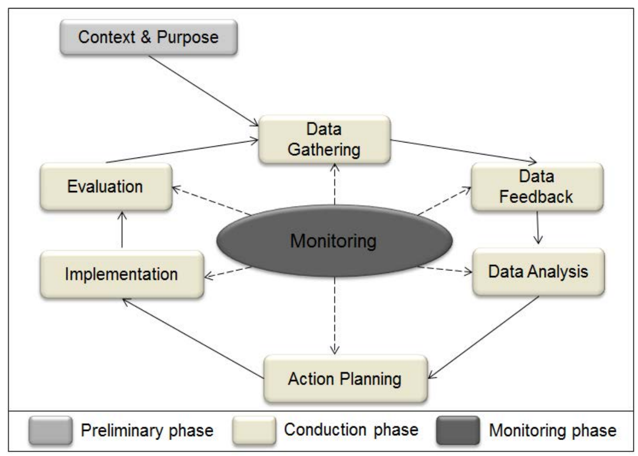


**Figure 1 Action-research lifecycle by Coughlan et al. [26]**

The research was executed during a period of two and a half years[1], composed by three cycles. The first two cycles [28] were named preliminary stage (cycle 0) and cycle 1.

The preliminary stage (Context and Purpose in Figure 1) had two premises: a) the project composed of computing teachers, with competences for the development of mobile apps, and students with technical, undergraduate, specialized, master, and doctorate

[1] The project had 2 classes of a "free course" targeted mainly to students of undergraduate courses, but also to masters and doctorate students.

backgrounds; and b) the teaching method used should be CBL and the physical environment should be adapted to include the use of this method. A reflection on those premises resulted on an initial plan: student selection, teachers preparation and execution of the project, that included a programming course, a challenge and its publication (Figure 2).

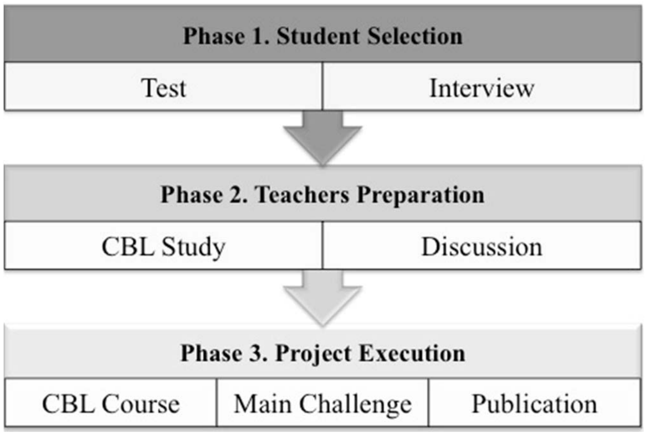


**Figure 2 Planned Phases**

In the second stage, cycle 1, the data collected and the analysys conducted in the previous cycle helped to identify problems and opportunities for improvement. As a consequence the last part of the original plan was changed to include more challenges and a support team for design, illustration, sound & music (Figure 4). The cycle 1 was executed and evaluated using surveys with students (78 answers from 91 students), interviews with teachers involved in this project (23 teachers from 8 brazilian universities). The analysis was conducted using numerical analysis of charts and thematic networks [29].
At the end of cycle 1, it was possible to cross-reference data collected during the project, such as the videos of students' reflections in the execution of the activities, and the evaluations undertaken by the students and teachers on the use of CBL. The following aspects were identified as the object of a change to cycle 2: improvement of students' selection based on the desired profile, gradual use of CBL, engaging students in a software development process, and orientation for the creation of startups.

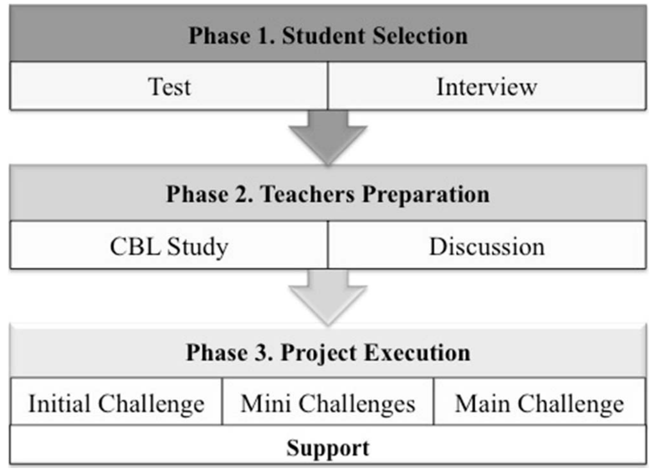


**Figure 4 Adjustments on the initial plan**

## 4 Cycle 2

Considering the data collected and evaluations of cycle 1, problems and opportunities for improvement were identified [28] and the teaching approach of IDEAS (Figure 3) was developed. This section describes "Action Planning," resulting from the findings, and its implementation and evaluation.

### 4.1 Cycle 2 Action Planning

In order to develop the IDEAS approach, we also considered the following requirements of a training course for mobile developers:

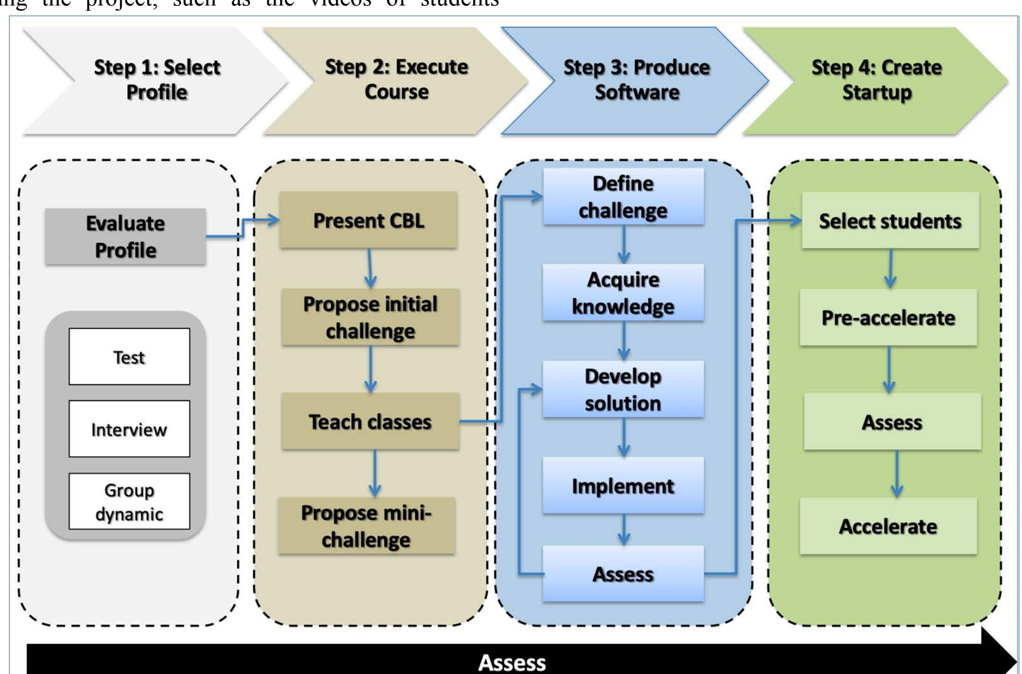




**Figure 3 IDEAS Approach**

i) use of the CBL method in a simplified manner during classes and in a complete manner in at least three real situations of software development; ii) teaching programming of the basic languages for iOS: Swift and Objective-C; iii) teaching app development for iOS; iv) gradual engagement of students in a software development process while learning programming; v) students' collaboration with the four development and publication support teams in the areas of usability and interface; musical tracks and sound effects; illustration, modeling, and animation; and entrepreneurship (startups); and vi) auxiliary training workshops to provide for the specific needs of student's projects, such as web systems and use of databases, among others.

Figure 3 presents an overview of the IDEAS approach and its four steps: Select Profile, Execute Course, Produce Software, and Create Startup. The following sections elaborate each of the steps that apply to the implementation of cycle 2.

## 4.2 Cycle 2 Implementation

This section describes the application of each step of the IDEAS approach presented in Figure 3.

4.2.*1* Select Profile. In cycle 1, we identified the need to improve the student selection process. We sought students with the following profile: logical reasoning ability, autonomy, commitment, proactivity, and collaboration. Therefore, the group dynamic was included in cycle 2.

The first part of the selection continued with the application of the **theoretical test**. The second part consisted of an **interview** with the selected candidates. The changes that occurred in this cycle were an increase of interview time (30 minutes) and adoption of the interview technique "Life Stories" [30]. This technique consisted of enumerating the positive and negative events mentioned by the interviewee, and primarily aims at identifying their personal interests and profile. Considering the group dynamic, the students were divided into teams of four and were given 36 hours to elaborate a proposal to solve a real-world problem that did not have an immediate solution. We proposed the following: to find alternatives to enable the continuity of print media in conjunction with the digital world. Each student was evaluated by at least 3 evaluators on the aspects: collaboration, leadership, proactivity, creativity, and autonomy. The ratings provided were stored in a mobile app designed for this purpose.

Approximately 900 candidates participated in the selection. Subsequent to the evaluation of the tests, 246 candidates were selected for the interview. The researcher conducted the interview and 220 candidates were selected to participate on another phase, a group dynamic. This process resulted in the selection of 100 students to be part of the subsequent class.

4.2.*2* Execute Course. Initially, the CBL method was presented in its simplified format. Successful case studies were presented to make the method attractive and motivating, highlighting its flexible dynamic, and collaborative characteristics.

Subsequently, the method was first applied to a quick task (usually less than 20 minutes), such as improving the audio of a noise-filled video, albeit difficult to execute for non-specialists. Then, a more extensive activity was proposed, though it did not involve the development of the software. The task consisted of developing an idea of a relevant and complete app, from the big idea to the "solution," which in this case was the production of a video presenting the final product. The purpose of the activity was to show the students that the model can be used to search for new ideas and convert them into products. After 4 to 5 weeks, the first mini challenge was issued.

The students were provided 2 days to set the first challenge, 2

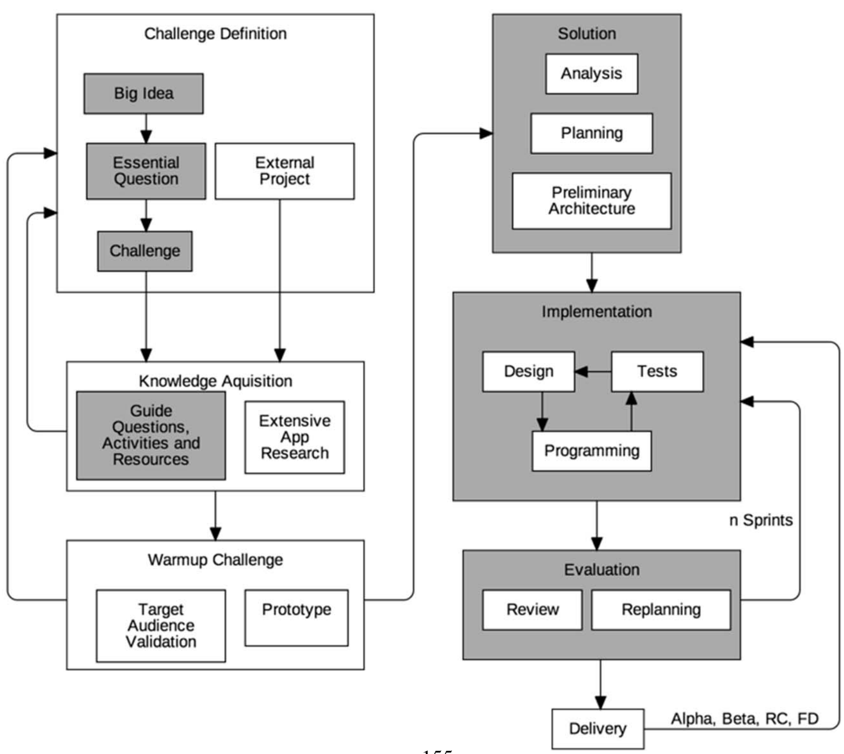




**Figure 5 Detailing of step 3: Produce software from Figure 2**

days to execute the guiding questions, activities, and resources; and on the 5th day, they were instructed by a teacher on planning, tasks division, and schedule. During this period, the students were observed and accompanied by teachers, considering the fact that although autonomy was encouraged, a critical problem could be often solved more quickly with the help of a teacher. At the end of 2 weeks, all the groups presented their products along with a promotional video and an individual reflection video. Subsequent to this, reinforcement classes were offered to students who performed less than expected in the weekly activities and the first mini challenge. The other topics of the schedule were superficially commented on, and soon afterward, the students were expected to conduct autonomous studies on a continuous basis with the goal of developing an app.

Subsequent to another 4 weeks, the students were expected to accomplish the 2nd mini challenge, where they were expected to develop a relevant app for publication. The goal was to consolidate software development skills under the constraints of time. This mini challenge follows the same pattern as the previous one, with a two-week deadline, in addition to one adjustment in presentation and publication.

4.2.*3* Produce Software. This step is aimed at completing the final challenge (i.e., developing and publishing a relevant app). According to Figure 5, an integration of the CBL method with a software process was carried out, aiming at addressing the challenges of the students in learning programming and to train students of different levels of knowledge. The dark-colored boxes in the figure represent the existing phases of the CBL method and the white-bottomed boxes represent activities of the integrated development process. The first four phases of the CBL were inserted into larger activities due to the need to perform other complementary tasks. The other three phases of the CBL (solution, implementation, and evaluation) originally have a very abstract definition and no detailing.

This approach provides an insight to the practice of software development with well-defined deadlines and using resources common to all, as students are engaged in a complex process of software development similar to a software factory in which multiple products are developed systematically. The resources were represented by content production support teams and teachers who act as project managers and technical consultants.

*4.2.3.1 Define Challenge.* The "Challenge Definition" can be completed by choosing an external project or using the initial phases of the CBL method (Big Idea, Essential Question, and Challenge). The external project can be chosen from the presentations of projects proposed by the professors external to the course, subsequent to the completion of a feasibility study of the project.

When the student prefers to seek their idea, they must begin with the “Big Idea.” In this activity, there is minimum intervention enabling the student to develop their perception and autonomy. Considering this, the students accomplish the activities based on their strategy and the theme definition, construction of a list of critical problems, and evaluation of the teacher. Subsequent to choosing the central theme, the student has to define the "Essential Question," which is a question of the selected theme. This question defines the challenge that the student is expected to face and solve through the development of an app.

*4.2.3.2 Acquire Knowledge.* This step is aimed at understanding the challenge and is conducted using the guiding questions, activities, and resources of the CBL method. Considering this, the students should identify a series of relevant guiding questions, answer then by performing the required guiding activities (reading, interviewing, prototyping, etc.) and undertake a critical evaluation if the answer is in sync with the guiding question. Extensive research is also conducted for similar apps. If an issue makes the project unfeasible, it will return to the previous activity ("Define Challenge"). This process is iterative, continuing until the goal of the guiding question is reached.

*4.2.3.3 Warmup Challenge.* The duration of this step is a maximum of two weeks. It aims to offer an opportunity for the students to evaluate the feasibility of their project and to better define the challenge with which they are expected to work. For instance, one of the proposed challenge was a game that uses simulation of physics, the prototype proposed was made using the framework Metal (not well documented at the time) and the main goal of the project was to identify if Metal was a viable technology

At this point, the students must decide whether there is an interest in creating a startup for their future product or not. If a startup is chosen, a canvas [31] should be created and a series of validations must be undertaken using the design thinking techniques [32], along with the entrepreneurship support team.

The other option is the development of a prototype of technical feasibility in which one or more critical technical points of the project are implemented and validated, based on the experience of the teachers. However, depending on the project, it may be necessary to accomplish both activities.

*4.2.3.4 Solution.* The solution step in the IDEAS approach represents the system analysis that results in the definition of an initial architecture and planning. In the analysis activity, the students develop a work breakdown structure [33] and then design a Gantt diagram as planned in the schedule.

In this step, we observed that the students faced challenges in the planning due to inexperience and diversity of the projects, primarily in the activities of scope definition and estimation of activities’ duration. These difficulties were solved through project management workshops.

*4.2.3.5 Implementation.* During implementation, an iterative cycle of designing, programming, and testing was defined, due to the need for a robust interface and dynamic usability requirements. The usability team provided the design of the interface. The students then programmed and tested the same on the representatives of the target audience. The results were delivered to the usability team and the cycle was repeated until there were no critical usability issues.

The decision on the form of communication and project management was the teams’ prerogative. A Scrum-like project follow-up framework created in a digital tool called "Trello" was suggested, which facilitated a follow-up by the responsible teacher. The teams were required to use the Git version control tool (website: https://git-scm.com/book/). Majority of the students selected the Bitbucket (website: https://www.atlassian.com), a hosting service for projects that require versioning control, working as a repository for the design.

For each project the main difficulties were tracked for future corrections. Each main difficulty encountered was followed by a corrective action for the current class and a preventive action for next classes and future projects. For instance, one of the difficulties was the interruption of the programming due to an update of Xcode that broke the code. The corrective action was to downgrade Xcode and the preventive action was to only update the software after a carefully reading of the new features and changes.

*4.2.3.6 Project control.* Only two teachers, both with business experience in project management, planned and controlled 47 projects of the class. The work was divided as follows: Teacher 1 monitored 18 projects (40% of the total) and managed the illustration support group “Academic Center for Games and Animation”(NAJA); Teacher 2 followed up on the remaining 29 projects.

A macro Gantt diagram was designed which contained all the teams. The follow-up was undertaken through weekly meetings of 15 minutes each, in which the participants discussed the following: percentage progress of the project, activities conducted in the previous week by a team member, difficulties encountered, subsequent week activities, attendance of the support centers, and identified risks. The meetings were recorded in a document called the "history log."

*4.2.3.7 Delivery.* During planning, four main delivery deadlines were defined:

- Alpha version: Term of one month. The purpose of this milestone is to provide a version with the main features of the app.
- Beta version: One and a half month from the delivery of the alpha version. All the planned main features should already be implemented. It is not necessary that all the design features must be applied to the software, but it is desirable that at least part of it is already working. Moreover, alternative streams for the same functionality need not be implemented.
- Release candidate (RC): The final month of development should be reserved for the applying the final design, polishing, testing, and bug fixes. At the end of the period, a candidate release version of the app should be made available for the final presentation to the committee.
- Final Delivery (FD): Final delivery of the app is expected one month after the RC. The app submitted is considered delivered and accepted by the store. In order to consider the project as completed, the students are required to organize the following delivery package: i) app source code; ii) materials developed by the support centers; iii) CBL documents, including a final reflection video; iv) folder with a description and screenshots of the app to be delivered to the store; and v) institutional video of the final product and final reflection video of the course.

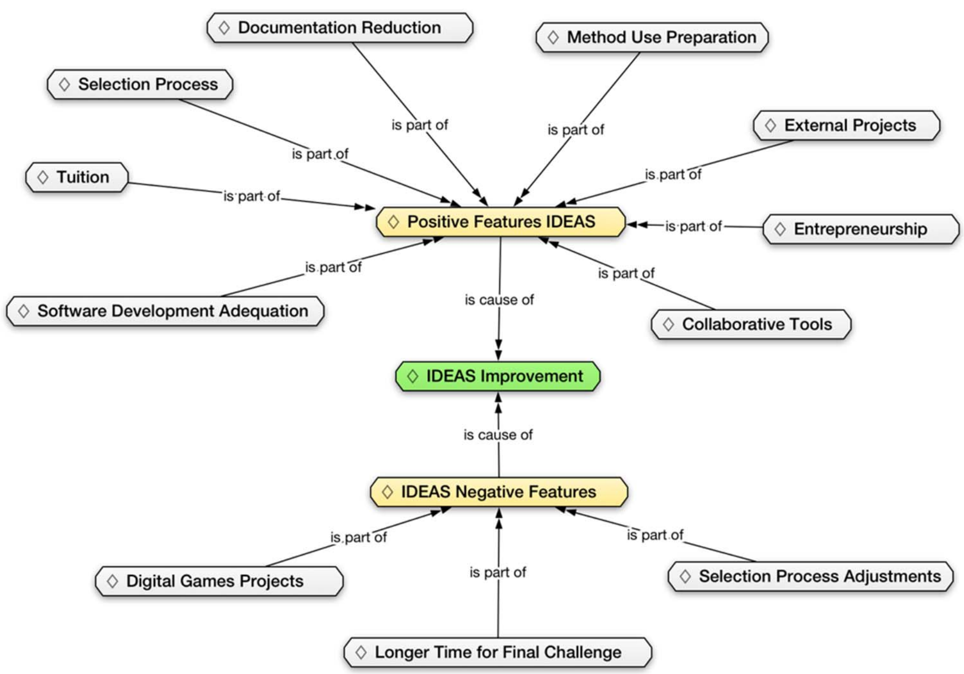


**Figure 6 Evaluation of the teachers using IDEAS**

Moreover, in addition to delivering the developed app to the store, all the teams should also gain the store's approval. This approval may require several modifications and resends until the app is approved. Subsequent to approval, the app is not required to be published and could be retained on a "on hold" status.

*4.2.3.8 Create Startup.* The startups were created in partnership with a company accelerator. An in-place accelerator with 44 unique workstations for startups of course graduates has been reserved. This step is composed of three parts: student selection, pre-acceleration, and acceleration following the definitions of Huijgevoort [34]. The criterion for selection was student performance during the course and software development. Exceptionally, as all the entrants had performed well during the course, all were selected for the pre-acceleration stage.

The duration of the pre-acceleration program is two months on an average, and it prepares the future entrepreneur to manage their business through mini-courses and individual guidance (mentoring). In this stage, the criterion of selection was to possess a minimally viable product developed. Owing to the fact that all the graduates of the course completed the course with their developed product, all were selected for evaluation. The criterion of evaluation for the acceleration stage is the achievement of the goals set by the mentors, indicating a commitment to the entrepreneurial culture. In this stage, which can last between six months and a year, the product is already in the market, used by customers, and generating engagement, attraction, and possibly revenue. This stage has two objectives: connection and leverage. The accelerator enables startups to enter a relationships chain, with the possibilities of a partnership with companies, an indication of customers and even investment, thus leveraging the business.

## 5 Cycle 2 Assesment

Several types of evaluations were conducted on the IDEAS approach during and at the end of cycle 2:

- Video-recorded follow-up interviews with nine students based on their position in the selection process: the first three, last three, and middle three. Each student was interviewed twice, once at the beginning and the other at the end of the theoretical part of the course (four months).
- Weekly reflections of all the students recorded in audio and video formats, aiming at checking the problems, and evaluating the specific suggestions and criticisms.
- Periodic reflections of teachers, written or recorded in audio and video formats, with similar goals to students' reflections.
- Final questionnaire with objective and subjective questions for all the students, aiming at verifying the same items evaluated in the first cycle with three more questions.
- Evaluation programming test for graduates who obtained a grade lower than 50% in the selection process test.

Final interview with objective and subjective questions with teachers, to verify the same items evaluated in the first cycle. Additional questions on the IDEAS approach were added.

### 5.1 General Evaluation

Cycle 2 presented direct results, which were measurable and positive results that were independent of specific evaluations. The main positive result was the percentage of graduates of the classes: 76.3% in the 1st group and 95% in the 2nd group. The 1st group excluded 18.2% of the students due to performance and commitment problems, while none of the students were excluded in the 2nd group. Considering that there was no change in the exclusion criteria, taking into account time, complexity of work, materials, and equipment provided, and the availability of teachers outside the class hours, it is prudent to consider that the changes made in the 2nd cycle, particularly those related to the selection process, were successful.

Another positive result was the increase in the percentage of projects approved by the appraisal committee. In the first group, 46.7% of the projects were evaluated as excellent (equal or superior to 80%), while a similar evaluation was provided to 88.9% of the projects in the second cycle. Engaging the students within a process of software development integrated to the CBL method possibly influenced the result.

A prominent result was in relation to the students selected for the pre-acceleration of their startups, where 14.2% of the first cycle graduates and 38.9% of the second cycle were selected. The inclusion of a support team for entrepreneurship, monetization workshops and target audience evaluation, a talk with one of the most respected names in the world of startups, and encouragement of commercial projects were responsible for this improvement.

Finally, the number of apps accepted and published in the app store was evaluated, where 41 were published from the first cycle and 84 from the second cycle. This difference can be explained by the improvement of the approach in general, but primarily by the changes in the selection process and integration of the software development process with the CBL method.

*5.1.1 Final Evaluation of the Students* A questionnaire[2] was distributed to the 95 students who finished the course, and 89 answered them. Among the respondents, 57.3% were novice students (less than a year of programming experience) and 19.1% were veterans (more than 3 years of programming experience).

**Table 1 Comparative of cycle assessments by the students**

| Questions | C1 | C2 |
|---|---|---|
| 1) How many times did you use the method? (> 3) | 71.8% | 89.9% |
| 2) Has the method helped in understanding the challenge? | 74.4% | 88.8% |
| 3) Has the method helped in developing a solution? | 70.5% | 82.0% |
| 4) Has the method helped in finding an innovative solution? | 48.7% | 49.4% |

[2] Most questions had answers from "Strongly disagree" to "Totally agree"

| | | |
|---|---|---|
| 5) What was the degree of method motivation? | 70.5% | 73.0% |
| 6) Do you intend to use the method in other situations? | 76.9% | 79.8% |
| 7) Has the method contributed in improving your skills as a programmer? (Novices) | 35.6% | 37.3% |
| 8) Have the guiding questions, activities, and resources helped in developing your apps? | 82.1% | 70.8% |

Table 1 presents a comparative summary of the results of the two cycle classes[3]. The rows represent the most significant questions for this comparison, and the columns signify the percentage of positive responses, where the 1st column shows the result of the 1st cycle and the second column shows the result of the 2th cycle. It shows that the result was better for seven out of the eight main points, and in three questions (questions 1, 2, and 3), the positive responses significantly exceeded the previous year (more than 10%). Only one of the items (question 8) presented a result inferior to the previous year. This could be primarily due to the inclusion of a requirements gathering stage within the software development process integrated with the CBL method.

In the space allocated for comments, some suggestions for improvement were cited, such as integration with a process of digital games development, increasing the time to completely accomplish the final challenge, and the need to improve certain aspects of the method.

### 5.1.2 Final Evaluation of the Teachers

11 teachers from 3 universities who met the following selection criteria participated in the interviews: completing their second iOS development class using the CBL.

The interview followed the guidelines same as that of the first cycle, except for the teachers of the institution where the IDEAS approach was applied, which answered some additional questions on applying the approach. Two thematic networks were generated, one referring to the teachers of the institution who used IDEAS and the other for the other teachers that did not use it. The aim was to undertake an evaluation of IDEAS .

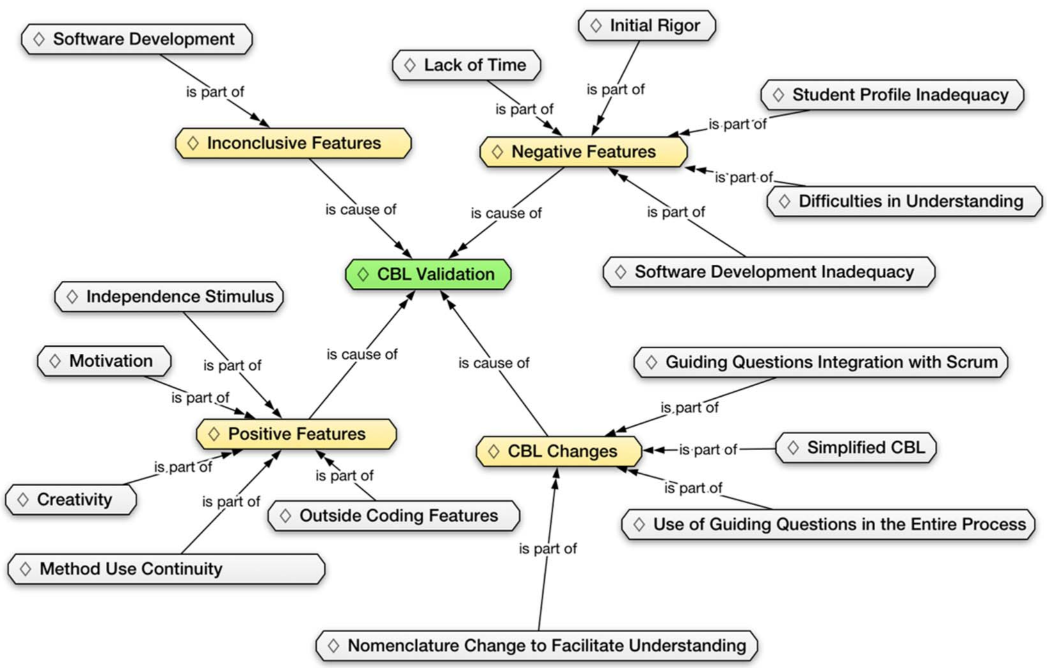


**Figure 7 Evaluation of the teachers of other universities**

Figure 6 shows the 1st thematic network, which represents an evaluation of the teachers who used IDEAS. The phrases in the figure represented the most frequent cited concepts by the interviewees. The analysis generated the main theme "IDEAS improvements," and the subtopics: negative and positive features of the IDEAS approach[4].

The most frequently cited **positive features** were those that underwent some type of modification in the first cycle. This direct

[3] The method referred by the questions in Table 1 is the CBL method

[4] In Figure 6: Tuition refers to constantly feedback given to the students by teachers. Software Development Adequation means the changes made to CBL in Cycle 2 as was shown in Section 4.2.

association can be easily explained by the fact that the teachers interviewed had taught in both the classes of the course. On the other hand, the negative features were as follows: a longer time for the final challenge, integration with a method of digital games development, and selection process adjustments.

The 2nd thematic network can be visualized in Figure 7 and represents the evaluation of teachers from other universities. The phrases in the figure represented the most frequent cited concepts by the interviewees. Several **negative and inconclusive features** were either the same or similar to those found in the analysis of the 1st cycle interviews. One relevant difference was the sub-theme "CBL changes," representing the changes made by these teachers in the execution of the succeeding class. Some of the prominent features were the use of a simplified version of CBL and the integration of the agile Scrum development method in the phase of guiding questions, activities, and resources, primarily to facilitate the management of projects.

## 6 Conclusions

This work presented the IDEAS approach, which is a distinct approach to support complex environments involving teaching, innovation, and software development for mobile devices, generating practical knowledge aimed at solving a specific problem, which are real and involve local interests. Although the goal was to develop mobile apps, there is no restriction for its use in other kinds of sofware development, like desktop apps or web apps.

We chose to adapt the CBL method in this approach, focusing on the development of a solution to a real problem, and the contents studied are developed from the needs of the proposed challenge. The research explored a complex and differentiated environment, in which CBL was adapted and applied specifically in special classes of mobile app development. The results show that the CBL method is appropriate for teaching programming in technology-rich learning environments [27].

Some original features of the IDEAS approach that are prominent are the creation of a differentiated selection process for its use in similar environments that go significantly beyond a programming knowledge test; the view of software development practice, as students are engaged in a complex process of software development similar to a software factory where several products are developed in a systematic manner with well defined deadlines and using resources common to all, from content production support teams to teachers who act as both, project managers and consultants on technical issues; and selection of the "makers" to create startups, facilitating the pre-acceleration process. In general, one of the stages of a pre-acceleration process is identifying the makers. The IDEAS approach already forms a professional with this profile, allowing for the elimination of the pre-acceleration stage. Another positive aspect reported was the reduction of drop-out and disapproval in a programming class, attributed to the assistance of teachers and students who were more experienced, in order to solve the challenges.

Considering future works, we intend to evaluate the IDEAS approach in other universities that conduct the training course for mobile app developers under the tutelage of different groups of teachers, enabling us to identify other problems and the respective improvements; evaluate the IDEAS approach under normal conditions of undergraduate courses (i.e., without scholarships, full-time teachers, and any other additional forms of motivation) only using the general equipment. The aim is to evaluate two main aspects. The first is to verify if the approach is effective and if it is possible to apply it in common situations arising from the economic reality of regions, where resources are not easily available to leverage a project of the magnitude of this research; and the second is to make adjustments to improve the innovation issues and programming skills. The innovation aspect is expected to be addressed in the third cycle through a study of the most popular and successful apps, and through innovation workshops conducted by a teacher who works in this area. The improvement of programming skills is expected to be addressed in the 4th cycle, which is expected to be planned at the end of the 3rd cycle.

## ACKNOWLEDGMENTS

We would like to thank the students and teachers who participated in this study. This work was done at Apple Developer Academy PUCPR.

## REFERENCES

[1] Lewis, G. A.; Nagappan, N.; Gray, J.; Rosenblum, D.; Muccini, H.; Shihab, E. Report of the 2013 ICSE: 1st International Workshop on Engineering Mobile-enabled Systems, 2013.

[2] Flora, H. K.; Chande, S. V. A review and analysis on mobile application development processes using agile methodologies. In: International Journal of Research in Computer Science, v. 3, n. 4, p. 9, 2013.

[3] Santos, A.; Sales, A.; Fernandes, P.; Nichols, M. 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. ACM, 2015.

[4] Naismith, L.; Lonsdale, P.; Vavoula, G.; Sharples, M. Literature review in mobile technologies and learning. Technical report, University of Birmingham, 2013.

[5] Hawthorne, Matthew J.; Perry, Dewayne E. Software engineering education in the era of outsourcing, distributed development, and open source software: Challenges and opportunities. In: Software Engineering Education in the Modern Age. Springer Berlin Heidelberg, 2005, pp. 166-185.

[6] Rosiene, C.P.; Rosiene, J.A. Flipping a Programming Course: the Good, the Bad, and the Ugly. In: Frontiers in Education Conference, 2015, El Paso, NM.Anais..., 2015, pp. 803-805.

[7] Kinnunen, P., & Malmi, L. (2005). Problems in Problem-Based Learning–Experiences, Analysis and Lessons Learned on an Introductory Programming Course. Informatics in Education - An International Journal, 4(2), pp. 193–214. Retrieved from http://www.ceeol.com/aspx/issuedetails.aspx?issueid=31a3afb2-d1d0-4c59-9692-0b5fb5be23a1&articleId=49c3dfc7-475b-4e17-b39b-6ca554648471.

[8] de Oliveira Fassbinder, A. G., Botelho, T. G., Martins, R. J., & Barbosa, E. F. Applying lipped classroom and problem-based learning in a CS1 course. In 2015 IEEE Frontiers in Education Conference (FIE) (pp. 1–7). IEEE. https://doi.org/10.1109/FIE.2015.7344223.

[9] Johnson, L. F.; Smith, R. S.; Smythe, J.; Varon, R. K. Challenge-based learning: An approach for our time. Technical report, The New Media Consortium, Austin, TX, 2009.

[10] Niemi, H. Active learning—a cultural change needed in teacher education and schools. In: Teaching and Teacher Education 18.7, 2002, pp. 763-780.

[11] Drew, V. e Mackie, L. Extending the constructs of active learning: Implications for teachers' pedagogy and practice. In: Curriculum Journal 22.4, 2011, pp. 451-467.

[12] Michael, J. Where's the evidence that active learning works? In: Advances in Physiology Education 30.4, 2006, pp. 159-167.

[13] Canaleta, X.; Vernet, D.; Vicent, L.; Montero, J. Master in teacher training: A real implementation of active learning. In: Computers in Human Behavior 31, 2014, pp. 651-658.

[14] Prince, M. Does active learning work? A review of the research. In: Journal of Engineering Education 93.3, 2004, pp. 223-231.

[15] Freeman, S.; Eddy, S.; Mcdonough, M.; Smith, M.; Okoroafor, N.; Hordt, H.; Wenderoth, M. Active learning increases student performance in science, engineering, and mathematics. In: Proceedings of the National Academy of Sciences 111.23, 2014, pp. 8410-8415.

[16] Bonwell, C.; Eison, J. Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports, Washington, 1991.

[17] Boud, D.; Feletti, G. Changing problem-based learning: introduction to second edition. In: BOUD, D.; FELETTI G. (Eds.). The challenge of problem-based learning. Londres: Kogan Page, 1999. pp. 1-14.

[18] Moalosi, R., Molokwane, S., Mothibedi, G. (2012). Using a Design-orientated Project to Attain Graduate Attributes, In: Design and Technology Education, v. 17.1, pp. 30-43.

[19] Tucker, Bill. The flipped classroom. In: Education Next, v. 12, n. 1, 2012.

[20] Nichols, M., Cator, K., & Torres, M. (2016). Challenge Based Learning Guide.

[22] Kinnunen, P.; Malmi, L. Problems in Problem-Based Learning-Experiences, Analysis and Lessons Learned on an Introductory Programming Course. In: Informatics in Education, v. 4, n. 2, 2005, pp. 193-214.

[23] Fassbinder, A.; Martins, R.; Botelho, T.; Barbosa, E. Applying flipped classroom and problem-based learning in a CS1 course. In: Frontiers in Education Conference, 2015, El Paso, NM.Anais..., 2015, pp. 803-805.

[24] Berbel, N. As metodologias ativas e a promoção da autonomia de estudantes. In: Semina: Ciências Sociais e Humanas 32.1, 2012, pp. 25-40.

[25] Scharff, C.; Verma, R. Scrum to support mobile application development projects in a just-in-time learning context. In: Proceedings of the 32nd ICSE, 2010, pp. 25-31.

[26] Coughlan, P.; Coughlan, D. Action research for operations management. International Journal of Operations and Production Management, 2002, v. 22, n. 2. Academic Research Library, 2002, pp. 220-240.

[27] Johnson, L. e Adams, S. Challenge Based Learning: The Report from the Implementation Project. Austin, Texas: The New Media Consortium, 2011, <challengebasedlearning.org/public/admin/docs/NMC_CBLi_Report_Oct_2011 > accessed 26 dez. 2018.

[28] Binder, F. V., Nichols, M., Reinehr, S., & Malucelli, A. (2017). Challenge Based Learning Applied to Mobile Software Development Teaching. In *Proceedings - 30th IEEE Conference on Software Engineering Education and Training, CSEE and T 2017* (Vol. 2017-Janua, pp. 57–64). https://doi.org/10.1109/CSEET.2017.19.

[29] Attride-Stirling, J. Thematic networks: An analytic tool for qualitative research. Qualitative Research, v. 1, n. 3, 2001, pp. 385-405.

[30] Atkinson, R. The Life Story in Interview. Handbook of Interview Research: Context & Method. P. 1a. Edição, Londres, Sage Publications. pp. 121-140.

[31] Osterwalder, A.; Pigneur, Y. Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. 1a edição. John Wiley and Sons, 2010. 288p.

[32] Polaine, A.; Lovlie, L.; Reason, B. Service Design: From Insight to Implementation. 1a edição. Rosenfeld Media LLC, 2013. 202 p.

[33] Project Management Institute, I. Um guia do conhecimento em gerenciamento de projetos (guia pmbok®). 5. ed. Newtown Square, Pennsylvania: Saraiva, 2013.

[34] Huijgevoort, T. The ‘Business Accelerator’: Just a Different Name for a Business Incubator. Bachelor Thesis, Utrecht School of Economics, Utrecht, The Netherlands, 2012.

[35] Vagrani, A., Kumar, N., & Ilavarasan, P. V. (2017). Decline in Mobile Application Life Cycle. In *Procedia Computer Science* (Vol. 122, pp. 957–964). Elsevier B.V. https://doi.org/10.1016/j.procs.2017.11.460