An Investigation of Influencing Factors when Teaching on Active Learning Environments

Nicolas Nascimento
PUCRS, School of Technology
Porto Alegre, RS, Brazil
nicolas.nascimento@pucrs.br

Alan R. Santos
PUCRS, School of Technology
Porto Alegre, RS, Brazil
alan.santos@pucrs.br

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

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

ABSTRACT

This paper presents an investigation on how students' background, previous experience and teams composition may influence the teaching process on active learning environments. Using a survey we were able to get data from students that participated in a two-year mobile application development course. We have found indicatives that the timeframe of an activity and the composition of the teams can have an impact students individual perception regarding the project. Moreover, our results also present indicatives that the number of students participating in a project may impact the students perception of projects.

CCS CONCEPTS

• **Social and professional topics → Software engineering education**; • **Applied computing → Interactive learning environments**; Collaborative learning.

KEYWORDS

Software Engineering, Learning Environment, Challenge Based Learning

ACM Reference Format:
Nicolas Nascimento, Alan R. Santos, Afonso Sales, and Rafael Chanin. 2019. An Investigation of Influencing Factors when Teaching on Active Learning Environments. In *XXXIII Brazilian Symposium on Software Engineering (SBES 2019), September 23–27, 2019, Salvador, Brazil.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3350768.3353819

1 INTRODUCTION

Active learning is a student-centered approach that fosters engagement, focusing on students rather than teachers, allowing learning to be performed collectively. Different studies reported active learning as a positive approach on mobile application development teaching and learning [2, 7, 8, 13]. In this sense, teaching high relevant skills such as Mobile Application Development (MAD) using an active learning approach has been shown to not only improve the learning experience, but also to produce solutions which directly address real-world problems [2, 10, 14].

Challenge Based Learning (CBL) [11] is one of the active methodologies which has been used when teaching such skills. CBL is a methodology created by educators and highly supported by *Apple Inc.* It is based on a three-stage approach, which allows students to: (i) **Engage** with a real-world problem; (ii) **Investigate** deeply about a theme; and (iii) **Act** on a feasible solution for the proposed problem.

As a teaching framework, CBL provides a high level of flexibility. Previous work has shown that it can be combined with Scrum, specifically for teaching MAD [13], and with Lean Startup and Customer Development concepts, supported by software development techniques [4]. Besides that, an investigation regarding CBL in active learning environments has been performed previously [6] for short iOS training courses.

In this context, we present in this paper an investigation of influencing factors when teaching MAD in an active learning environment using CBL. The investigation was performed on a two-year course that teaches MAD to undergraduate students. The factors investigated were *previous working experience*, *team size* and *project time duration* and their influence on student perception about their projects. After discussing and exploring the investigation, some results indicatives are provided. Our results present indicatives that these factors may impact the perception of students.

The remainder of this paper is organized as follows. Section 2 briefly contextualizes important concepts. In Section 3 we describe the course being studied and, in Section 4, we present the scientific approach used for evaluation and analysis of the collected data. Following this, Section 5 depicts the results and Section 6 presents a brief discussion and highlights some important findings from this study. Section 7 describes some threats to the study. Finally, Section 8 concludes the paper with some final thoughts and future work.

SBES 2019, September 23–27, 2019, Salvador, Brazil

ACM ISBN 978-1-4503-7651-8/19/09.
https://doi.org/10.1145/3350768.3353819

2 BACKGROUND

The context that will be further explained in Section 3 have already being explored by other studies [5, 6]. These two studies focused more on how the CBL methodology fits into the environment. In this study, on the other hand, we are interested in exploring human factors that can improve (or jeopardise) the students' learning process.

2.1 Active-Learning

Teaching students can be done in multiple ways. Traditionally, learning is mostly based on lectures, a teacher-centered approach which usually provides low levels of interaction. On the other hand, active learning is an approach that proposes high levels of interaction and stimulates students to perform not only low-order cognitive tasks, such as reading and writing, but also high-order ones, such as debating, analysing and decision making [2, 7, 8].

2.2 Challenge Based Learning

There are several active learning methodologies that have been used in an educational setting. Problem Based Learning, Project Based Learning, Task Based Learning and Challenge Based Learning are just a few examples of these frameworks. *"The foundations of experiential learning can be found within the history of most cultures, but were formally organized and presented by David Kolb drawing heavily on the works of John Dewey and Jean Piaget"* [13]. Challenge Based Learning (CBL) [12] is a learning framework based on solving real world challenges.

CBL was developed by educators working with *Apple Inc.* [11] and has been implemented both in educational and corporate environments. From an education perspective, students acquire knowledge by working on open-ended problems. In the CBL process, students, tutors and other stakeholders work together as active collaborators. In addition, the focus is not on the final deliverable, but rather on the whole process. In this sense, students must reflect from time to time on their learning evolution as shown in the framework presented in Figure 1.

The CBL process begins with the definition of a *big idea*, which is a broad concept that can be explored in several ways. The big idea has to be engaging and important to students. Once the big idea is chosen and the *essential question* is created, the *challenge* is defined. From this point, students must come up with the *guiding questions* and *guiding activities and resources*, which will guide them to develop a successful solution. The next step is *analysis*, which will set the foundation for the definition of the *solution*. Once the solution is agreed upon, the *implementation* begins. Finally, *evaluation* is undertaken in order to check out the whole process and verify if the solution can be refined.

Those CBL [12] framework stages (*Engage*, *Investigate* and *Act*) have their own set of activities as the following:

- ***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* which students want to explore;
  Challenge: a call to action derived from the essential question. It should be actionable and exciting.

Figure 1: Challenge Based Learning Framework [12].

- ***Investigate***
  Guiding Questions: all questions related to the challenge. Includes everything that needs to be learned;
  Guiding Activities and Resources: the list of activities and resources that support students to pursue the challenge;
  Analysis: sets the foundation to develop a solution to the challenge.
- ***Act***
  Solution Development: based on findings from the previous steps, a solution is implemented;
  Evaluation: verifies if the solution has addressed the challenge or if it needs refinement.

The CBL framework is flexible. In this sense, a wide variety of topics can be taught through CBL, including MAD, and it can be integrated with other frameworks, *e.g.*, Scrum, Lean, *etc.*

2.3 Related Work

Santos *et al.* [13] conducted an empirical study about the combination of CBL and Scrum. The main focus of the study is the students perception of CBL and Scrum and its effectiveness when used for learning of MAD. Although the results indicate that combining CBL and Scrum can be highly effective in the learning process, the study does not explore other important factors for the success of the implementation, such as the background of students.

Chanin *et al.* [5] conducted a case study on the perception of students regarding whether CBL could help to improve their software engineering skills when working in software startups. Even though the findings were positive, the study lacks a further investigation of correlating factors for the success of CBL in the learning environment.

In this context, given the gaps these studies presented, there is an opportunity to understand what other factors can influence students perception when learning on active learning environments.

3 THE COURSE

This study focuses on a two-year mobile development course that teaches iOS, tvOS and watchOS development to undergraduate students. All learning in the course follows the CBL principles. In the course, students are introduced to the development ecosystem of Apple platforms and learn by working on real world problems. Besides that, students can choose to focus on development or design aspects of mobile application development. Students in the course are expected to dedicate 20 hours per week at course activities.

During the course, as CBL proposes, students learn through challenges. As described by Nichols *et al.* [12], the challenges can be classified as:

- **Nano-Challenge**: These challenges are shorter in length, focused on a particular content area or skill, have tight boundaries and are guided by the instructor. Both *Big Idea* and *Essential Question* are provided to the students. The process includes some level of investigation, but at a lower level of intensity and often stop short of implementation with an external audience;
- **Mini-Challenge**: These challenges have a longer duration (2-4 weeks) and allows learners to start with a *Big Idea* and work using the entire framework. The research depth and the reach of their solutions increase and the focus can be content specific or multidisciplinary. Mini-Challenges are good for intense learning experiences that stretch learners and prepare them for longer challenges;
- **Standard-Challenge**: These challenges are the longest (one month and longer) specified in framework and allow considerable latitude for the learners. Working together, learners identify and investigate *Big Ideas*, develop *Challenges*, do extensive investigation across multiple disciplines and take full ownership of the process. The framework is used from beginning to end, including implementation and evaluation of the solution in an authentic setting.

Due to the variety in length of the challenges, students work on a larger number of short challenges (**Nano-Challenge**) than in longer challenges (**Standard-Challenge**).

4 METHODOLOGY

This study was conducted through survey for data collection and classification [3]. Survey is appropriate when the focus of interest is on what is happening or how and why something is happening and also applies when it is not possible to control dependent and independent variables [3]. This study aims to answer the following research questions:

- **RQ1**: *"How can previous working experience change perceptions of students on active learning environment?"*
- **RQ2**: *"What is the influence of team size in student perceptions on active learning environment?"*
- **RQ3**: *"What is the influence of project duration in student perception on active learning environment?"*

In order to answer these questions, the following steps were undertaken.

4.1 Data Collection & Analysis

A survey was conducted after the end of the two-year program in order to measure variables related to the individual perception that each student had regarding all developed projects.

4.1.1 Survey Protocol. The goal of the survey conducted in this study was to identify relevant aspects that influence the project development. The sample population (47 students) was composed of undergraduate students in a mobile application development program, who were chosen using the convenience criteria due to the fact that participants were selected for their availability. The sample population size was defined using the higher number of available people to participate in this study.

In order to avoid students from associating the questions presented in the survey with their individual evaluation, each student was told about the purpose of the research and about the zero-impact their answers had on their evaluation. In the survey, students were asked to individually evaluate the projects they had worked on while attending the course. While answering the survey, a researcher was available to remove any doubts regarding the survey. In addition, projects which were shorter than one week were not considered in the study.

Although the sample population is 47 students, the number of teams of students working on different projects is 90. This number ended up being larger than the number of students in the study (47) due to the multiple team configurations each student had participated.

During evaluation of projects, students could take into consideration any aspect regarding the project they found relevant. Students had to provide a grade from 1 to 5 (where 1 is the lowest and 5 the highest), using the same criteria internally applied by the instructors.

Throughout the course, instructors collected data regarding each project students have worked on. This data was obtained by averaging the instructors evaluation of a project performed by the students. This evaluation ranged from 1 to 5, following a Likert scale. The criteria used for this evaluation was:

(1) Terrible performance;
(2) Poor performance;
(3) Average performance;
(4) Good performance;
(5) Amazing performance.

We have also collected students demographic data including age, gender, role (developer or designer) and previous work experience. Regarding previous working experience (prior to the course), students were classified in three categories:

(1) *Full-time employment* - A student who has experience of working professionally in a software development company as a full-time software developer.
(2) *Internship* - A student that has either worked professionally in a software development company as an intern software developer or worked with software development in extra-curricular projects with focus on software development.

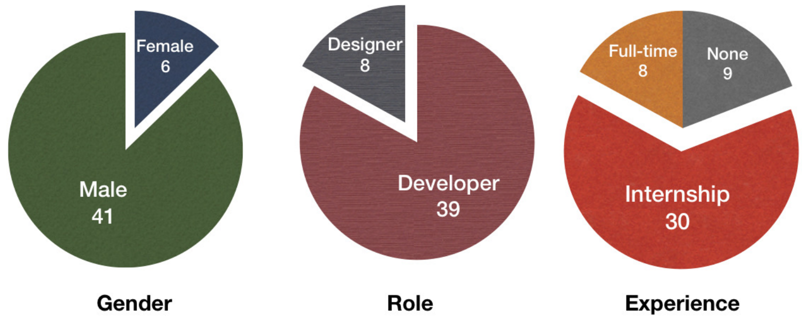

Figure 2: Demography of students

(3) *None* - A student who has no experience developing software or has only had experiences which did not fit the criteria for above categories.

Once data was collected, grouping and categorization of the information were performed. All data was stored on an Airtable database[1].

In order to perform the analysis, data from the survey, demography and project evaluation (both from the students and the instructors) were analyzed in a quantitative manner.

5 RESULTS

After data collection, data analysis was performed using triangulation. From a demography standpoint, the average age was 21.5 years old, with a standard deviation of 3.18 years. Figure 2 presents the demography of students regarding gender, role and previous students experience.

5.1 Projects Score

As an initial analysis, Figure 3 presents the score of 9 projects developed by the students. In order to allow comparison to be performed, for each project, the left (solid) bar represents the average rating provided by the students in the survey and the right (patterned) bar represents the average rating provided by the instructors (for the same project).

Figure 3 demonstrates that the average instructors score regarding projects was at least 1 point higher than the students individual perception, almost reaching 2 points in some cases. Reasons behind this can be related to the initial expectation of students for a project outcome to be much higher than what is actually achieved after finishing it. Another possible reason is that students focus exclusively on the final results and end up not considering the learning experience of the project.

In addition, reasons for these differences could also be contextual to the period of time in which activities were carried out. As the course in which students were enrolled was two years long, a project may have been impacted by external factors, such as schedule updates. Due lack of information regarding that matter for the collected data, we were unable to analyse this relation.

From this initial set of data, we continued our analysis by performing multiple grouping strategies to the collected data. Each grouping strategy provided insights regarding the research questions of this study.

[1]https://airtable.com/

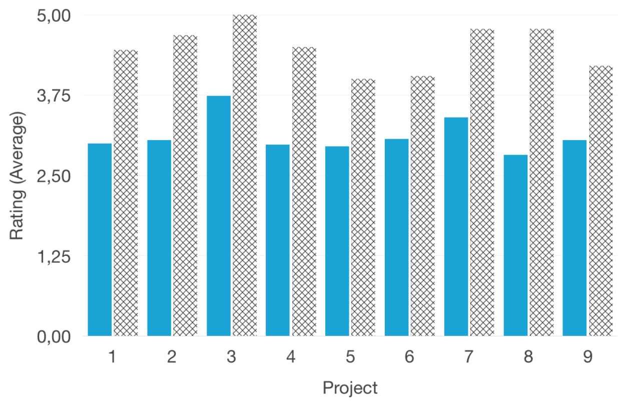

Figure 3: Projects Score

5.2 Student Experience

The first grouping was related to **RQ1** and it was based on the previous working experience of students. In order to perform this, we have assigned each student a score associated with his/her previous working experience category (*Full-time*, *Internship* or *None*). This score was based on a system of points where a student could be assigned:

- **10 points** - If the previous working experience category of the student was *Full-time*.
- **5 points** - If the previous working experience category of the student was *Internship*.
- **0 points** - If the previous working experience category of the student was *None*.

After this step, each of the 90 teams was assigned a team score. Teams score was obtained by averaging individual scores of students participating in the team. Finally, we categorized teams in five categories as follows:

- ***Novice***: if the average score of the team was under 2 (excluding 2);
- ***Beginner***: if the average score of the team was between 2 and 4 points (including 2, but excluding 4);

- ***Intermediate***: if the average score of the team was between 4 and 6 points (including 4, but excluding 6);
- ***Advanced***: if the average score of the team was between 6 and 8 points (including 6, but excluding 8);
- ***Expert***: if the average score of the team was above 8 points.

Table 1 presents the results from the survey after grouping teams based on this criteria.

Table 1: Survey results grouped by category

Category	# Teams	Average	Std. Deviation
Novice	2	2.38	0.72
Beginner	19	3.11	0.74
Intermediate	40	2.88	0.71
Advanced	28	3.47	0.66
Expert	1	3.67	0.47

The obtained results present indicatives that the perception of students could be associated with the previous working experience from the team members. By not considering the *Novice* and *Expert* categories, which accounted for less than 4% of teams, the highest average rating was given to teams that were categorized as *Advanced*, followed by *Beginner* and *Intermediate* teams. Although these results are only indicatives, reasons can be related to a higher level of difficult of student to cooperate when the knowledge gap is large or when students face team activities in a competitive manner [9].

5.3 Team Size

In relation to **RQ2**, we also performed data analysis grouping all teams based on their size (*e.g.*, number of students). In this context, it is worthy to mention that students worked in teams for all projects. In those teams, they had to work at least in pairs. After this grouping strategy was performed, the results from the survey are presented in Figure 4.

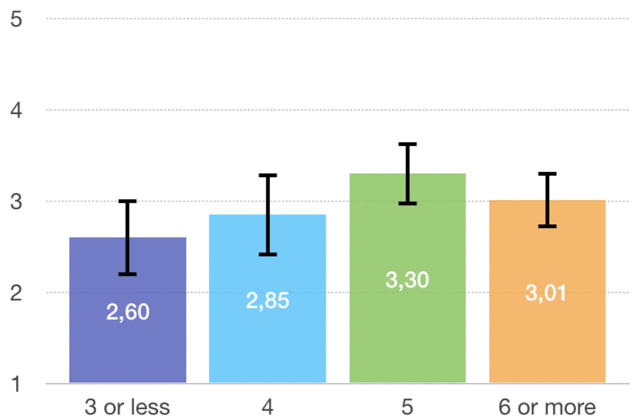

Figure 4: Project Score x Team Size

Regarding the number of teams for each category, the grouping resulted in seven (7) teams with 3 or less students, eighteen (18) teams with 4 students, fifty (50) teams with 5 students and fifteen (15) teams with 6 or more students.

Through this perspective, it is possible to visualize that the perception of students is somehow influenced by the number of components in their group. By considering the average rating of projects, teams with 5 components had the highest average rating among students, followed by teams with "6 or more", "4", and "3 or less".

5.4 Project Duration

Finally, regarding **RQ3** and as a final analysis, we have grouped the results from the survey considering the different duration of projects. In this sense, following the types of challenges proposed by the CBL framework, teams were grouped based on the duration of the challenge to which they were associated.

This grouping have resulted in three categories: 1 week (*e.g.*, *Nano-challenges*) teams, 4-6 weeks (*e.g.*, *Mini-challenges*) teams and 26 weeks (*e.g.*, *Standard-challenges*) teams.

After applying this grouping strategy, we obtained forty-one (41) teams which were from *Nano-challenges*, forty (40) teams which were from *Mini-challenges* and nine (9) teams which were from *Standard-challenges*. In this context, the average score of teams is presented in Figure 5.

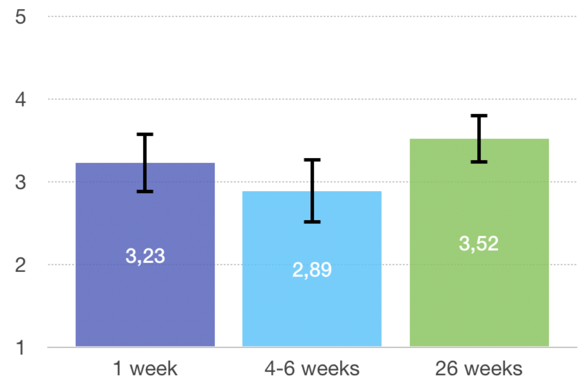

Figure 5: Projects Score x Projects Duration

From this perspective, results seem to indicate that project duration may somehow influence the perspective of students. *Standard-challenges* had a better average score when compared to both *Mini-Challenges* and *Nano-Challenges*. As students are always aware of the duration of projects, some reasons behind this can be related to expectation for a project.

On short-term projects, students may have lower expectations and thus tend to be more optimistic about the final result. For intermediate projects, (*e.g.*, *Mini-Challenges*), students may have higher expectations and thus get frustrated with the final result. For longer projects, the same expectation process can happen to students, with the difference that the longer period of time allows corrections to be made.

6 DISCUSSION

During the course, students worked together in teams and used CBL as the learning framework and Scrum as the development framework, since the combination of these frameworks has been successfully combined and tested in a previous work [13]. In this sense, as most activities in the course required the development of small releases of software, it is fair to associate teams of students to a software development team that uses Scrum. Following this idea, the Scrum Guide [15] defines that the development team should be composed of not less than three (3) and not more than nine (9) participants. This is aligned with the results from the survey conducted with the students, where the optimal team size was five (5) people.

When the results were grouped by students' expertise, the *"intermediate"* level had the lowest rating on average. These results are in agreement with existing literature [1]. Acuña *et al.* [1] demonstrated that multiple personality traits can also have significant impact on team performance and satisfaction, such as team agreeableness impacting team cohesion. As this personality trait was not extracted from the survey, it might have had impact on students perception.

In terms of projects' duration performed by the students, the highest average rating was given to projects with the longest duration. Previous research [16] pointed out that recognition, which is more feasible to happen on long-term projects, may play a relevant role in software development projects.

7 THREATS TO VALIDITY

Although results obtained by this study were mostly supported by previous work presented by the literature, some threats to validity are present and may diminish the generalization of the findings reported. Firstly, assessment of projects developed by students was performed at multiple times during the course. In this sense, as students were cumulatively introduced to new concepts and had the opportunity to work in more projects, their familiarity with development practices improved. The assessment conducted by the instructors, which was subjective, may have suffered influence where projects developed later in time are more likely to present better results.

The expertise was evaluated considering the working experience students had before undertaking the course. In this sense, it does not consider the learning that occurs on during projects development and activities in the course. It is plausible, for example, to assume that a student who had no previous experience might outperform a student with a full-time job experience during development of projects. Furthermore, a student with no previous experience may put more effort in learning on projects development and surpass students with more experience.

Moreover, the survey was conducted at the end of the second year of training. This may present a threat as students had to remember about projects they worked on the previous year. Even with the development artefacts available, students might not have been able to remember key occurrences of the project.

8 CONCLUSION

This paper presented results from a survey conducted with students on a mobile application development course environment regarding their individual projects perceptions. These results were analysed through data analysis and triangulation, and they were also compared to existing literature about similar topics.

Our preliminary results demonstrated indicatives that there is an assessment difference among students and instructors perceptions. We have found at least 1 point evaluation difference (in a scale of 5 points). Students self-assessment tends to result in lower rating. Also, it was shown that project's duration, teams composition (regarding previous work experience) and teams size play important roles in the students' individual perceptions.

It is worthy to mention that the survey was conducted at the end of the two-year program and some of the questions required students to remember projects they had developed more than one year ago. This indicative can be considered a study limitation and will require future work to explore it.

Regarding future work we intend to repeat this survey in future courses so we can further investigate whether our findings are concise. However, we will gather information from students throughout the whole 2-year process so we can avoid the aforementioned limitation.

REFERENCES

[1] S. T. Acuña, M. Gómez, and N. Juristo. 2009. How do personality, team processes and task characteristics relate to job satisfaction and software quality? *Information and Software Technology* 51, 3 (2009), 627–639.

[2] K. Ahmad and P. Gestwicki. 2013. Studio-based Learning and App Inventor for Android in an Introductory CS Course for Non-majors. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE '13)*. ACM, New York, NY, USA, 287–292.

[3] E. Babbie. 2005. *Survey research methods.* UFMG.

[4] Rafael Chanin, Afonso Sales, Leandro Bento Pompermaier, and Rafael Prikladnicki. 2018. Challenge based startup learning: a framework to teach software startup. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE 2018, Larnaca, Cyprus, July 02-04, 2018.* 266–271. https://doi.org/10.1145/3197091.3197122

[5] R. Chanin, A. Sales, A. R. Santos, L. Pompermaier, and R. Prikladnicki. 2018. A Collaborative Approach to Teaching Software Startups: Findings from a Study Using Challenge Based Learning. In *Proc. of the 11th Inter. Workshop on Cooperative and Human Aspects of Software Engineering (CHASE '18)*. ACM, New York, NY, USA, 9–12.

[6] R. Chanin, A. R. Santos, N. Nascimento, A. Sales, L. Pompermaier, and R. Prikladnicki. 2018. Integrating Challenge Based Learning Into a Smart Learning Environment: Findings From a Mobile Application Development Course (P). In *The 30th International Conference on Software Engineering and Knowledge Engineering, Redwood City, California, USA, July 1-3, 2018.* 704–703.

[7] M. Fetaji and B. Fetaji. 2009. Analyses of mobile learning software solution in education using the task based learning approach. In *Information Technology Interfaces, 2009. ITI '09. Proc. of the ITI 2009 31st Int. Conf. on.* 373–378.

[8] P. Gestwicki and K. Ahmad. 2011. App Inventor for Android with Studio-based Learning. *Journal of Computing Sciences in Colleges* 27, 1 (Oct. 2011), 55–63.

[9] Anita Lie. 2002. Cooperative learning.

[10] V. Matos and R. Grasser. 2010. Building Applications for the Android OS Mobile Platform: A Primer and Course Materials. *Journal of Computing Sciences in Colleges* 26, 1 (Oct. 2010), 23–29.

[11] 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

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

[13] 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.

[14] C. Scharff, A. Wasilewska, J. Wong, M. Bousso, I. Ndiaye, and C. Sarr. 2009. A model for teaching mobile application development for social changes: Implementation and lessons learned in senegal. In *Int. Multiconf. on Computer Science and Information Technology, 2009. IMCSIT '09.* Wisla, Poland, 383–389.

[15] K. Schwaber and J. Sutherland. 2011. The scrum guide. *Scrum Alliance* 21 (2011).

[16] O. Zwikael and E. Unger-Aviram. 2010. HRM in project groups: The effect of project duration on team development effectiveness. *International Journal of Project Management* 28, 5 (2010), 413–421.