Integrating Project Based Learning and Project Management for Software Engineering Teaching: An Experience Report

Maria Lydia Fioravanti, Bruno Sena, Leo Natan Paschoal, Laíza R. Silva, Ana P. Allian, Elisa Y. Nakagawa, Simone R. S. Souza, Seiji Isotani, Ellen F. Barbosa
Institute of Mathematics and Computer Science (ICMC) - University of São Paulo (USP) - São Carlos/SP, Brazil
{mlfioravanti,brunosena,paschoalln,laizaribeiro,ana.allian}@usp.br,{elisa,srocio,sisotani,francine}@icmc.usp.br

ABSTRACT

Software Engineering (SE) is an important topic to be taught in Computer Science courses. However, teaching of theoretical concepts with no link to their practical applications or no examples in the student's context may discourage learning, justifying why teaching and learning are great challenges of education in universities. In attempt to bridge such gap, several approaches have been proposed and applied to improve teaching and learning SE such as project based learning (PBL), a well-known approach already applied to teach SE. Nevertheless, there's a lack of understanding about how to better apply PBL and how to take advantage of this approach, for future use. There is also a lack of experience report describing how to use, its challenges and difficulties, what could be hampering to widely adopt it. We present our experience applying a PBL approach combined with project management to create an environment considering aspects such as dealing with managers and real stakeholders. The goal is to bring students closer to the reality of developing a software project in the business context. Our experience indicates positive results on the adoption of a PBL approach. In general, students were enthusiastic and positive about the use of this approach, the presence of a manager and the importance of using real-world problems with real stakeholders.

CCS CONCEPTS

• **Social professional topics → Software engineering education**;

KEYWORDS

SE Education, PBL, Software Project

ACM Reference Format:
Maria Lydia Fioravanti, Bruno Sena, Leo Natan Paschoal, Laíza R. Silva, Ana P. Allian, Elisa Y. Nakagawa, Simone R. S. Souza, Seiji Isotani, Ellen F. Barbosa. 2018. Integrating Project Based Learning and Project Management for Software Engineering Teaching: An Experience Report. In *SIGCSE '18: The 49th ACM Technical Symposium on Computer Science Education, Feb. 21–24, 2018, Baltimore, MD, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3159450.3159599

SIGCSE '18, Feb. 21–24, 2018, Baltimore, MD, USA

ACM ISBN 978-1-4503-5103-4/18/02…$15.00
https://doi.org/10.1145/3159450.3159599

1 INTRODUCTION

IEEE [9] defines that "*Software Engineering (SE) is the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is, the application of engineering to software*". According to the curriculum guidelines for undergraduate degree programs in several computing courses, SE is a knowledge area that must be addressed in higher education. In Computer Science courses, it is a whole knowledge area [1]; in Computer Engineering courses, it is thought as a topic [3]. A particular case is the Software Engineering degree program that is all dedicated to this area [2].

Even knowing how important is to teach Software Engineering, teaching and learning are great challenges, especially at the university. Besides transmitting information, teachers must deal with different students' learning styles, needs and strategies [14]. Another issue that may discourage graduate and undergraduate students is the disassociation between theory and practice [12]. The teaching of basic concepts and theoretical foundations with no link to their practical applications or no examples in the students' contexts is usual in computing courses. In an attempt to bridge this gap, several resources and approaches have been proposed and adopted to improve learning and teaching in computing courses. Particularly, we highlight PBL approaches, which aim to bring students closer to practice by means of a real project. In this sense, this paper presents our experience when teaching Software Engineering based on a PBL approach in a Software Engineering course of a Computer Science program. It is important to mention that our approach differs from those commonly used because it considers a very important element when it comes to software projects: the project manager. The focus of this paper is not presenting the approach itself, but tboth graduate and undergraduate students' perceptions of its use such as the challenges that were faced in this process and the lessons learned.

The paper is structured as follows: Section 2 presents the background; Section 3 briefly describes our PBL approach; Section 4 discusses the survey conducted for the approach evaluation by undergraduate students and also shows the opinions of graduate students towards the adopted approach. Section 5 addresses the lessons learned and challenges faced. Finally, conclusions and perspectives for future work are drawn in Section 6.

2 BACKGROUND

Several studies have shown how difficult is to share Software Engineering knowledge [4, 6, 7, 11]. SE has been traditionally taught through lectures. Blackboards and slide presentations are used as a support to the explanations. In addition, students should take part in a fictitious project, which has little or no connection to the practice of software engineering. However, these are very static methods, since the teacher will follow the class plan rigorously, with little room for changes.

There are several approaches other than the traditional one to teach SE and also motivate students to engage in the learning process, that is, become active. Considering how complex is to teach SE only through readings and projects, several authors have proposed approaches to address this issue. For instance, educational card game[4], case study, group work, experiment and cooperative education programs [6]. According to Bonwell and Eison [5], active learning can be essentially defined as "*students doing things and thinking about what they are doing*". The aim of active learning is to provide opportunities for learners to critically think about content through a range of activities that help prepare learners for the challenges of professional situations.

In this sense, PBL involves deep learning, as it focuses on real-world problems and challenges and relies on problem solving, decision making and investigative skills. These characteristics are suitable for improving the teaching of Software Engineering. Learning SE and how to apply your approaches throughout a software project can be a key factor in being successful. However, the success of a project requires more than knowing SE and how to apply it, it requires input from a variety of groups including the client, the project team, the project manager, the organization, the producer and the end user. Each party has a role in defining and determining success and they all have specific tasks and responsibilities that they must fulfill in order to achieve success. The project team will shape the implementation of the project. It is important for the team to employ the correct management techniques to ensure that planning, controlling and communication systems are all in place. Without these systems the coordination and control of all individuals and resources within the team is difficult [13].

Even though the use of a PBL approach brings students closer to reality, it is important to consider other aspects to successfully conduct a software project. In this sense, we have to consider a key factor in software projects: the project manager. The project manager is the individual responsible for delivering the project. The individual leads and manages the project team, with authority and responsibility from the project board, to run the project on a day-to-day basis. Kerzner [10] states that "*the major factor for the successful implementation of project management is that the project manager and team become the focal point of integrative responsibility*".

3 OVERVIEW OF OUR PBL APPROACH FOR SOFTWARE ENGINEERING

In an attempt to bridge the gaps previously mentioned, we adopted a PBL approach for teaching Software Engineering. The lectures were interspersed with project lessons, so the students learned the theory and then applied it to the project. The project comprised four stages, each one being related to the content taught in the classes prior to its beginning.

The main idea of the project was to bring students closer to the reality of developing a software project in the business context. This real-world scenario also included real clients, who were responsible for detailing the specifications of the desired systems. The differential of our approach with respect to those usually adopted is the presence of a graduate student as a project manager, which we believe is a key factor to improve the undergraduate students' learning. Each graduate student had to manage a whole team composed of 8-10 undergraduate students. They were free to choose any process model for conducting the project, and they could also choose any communication tool, configuration management tool and project management tool.

The focus of the adopted approach was to combine theory and practice of Software Engineering and Project Management, comprising the following main activities: requirements elicitation, modeling, design and academic software development management in a five-month SE course at University of São Paulo. Two professors and one teaching assistant were responsible for the theoretical lectures and course management, and 10 graduate students and 99 undergraduate students were supposed to perform the proposed activities of the PBL approach.

The students were divided into two groups, composed of five teams. Each team consisted of 8-10 undergraduate students and one graduate student and was responsible for the development of one academic software.

Although in this paper we deal with the dynamics of the group in which the authors participated, the dynamics of the other groups was analogous. Each graduate student had the role of manager of a team, being responsible for defining and managing the project schedule, allocating the necessary resources for each activity, ensuring the use of the process model adopted by the team, and so on. Following, we present the stages of the project, explaining the dynamics of each one.

Stage 1: Requirements Elicitation

Initially, teams of undergraduate students were created by teachers based on the students' previous knowledge and their profile. Each manager interviewed the teams and set an order of priority for the groups they would like to work with. Finally, the groups were assigned to managers according to their priorities and the theme was assigned randomly.

The first stage of the project concerned the requirements elicitation and analysis activities. An interview with the stakeholders was conduct to allow students to extract the system's requirements in detail. Thereafter, the students could contact the stakeholders to solve any raised questions during the elicitation process. The deliverable artifact of this stage was a Requirements Document that should specify all the system's details. It should be handed in six weeks after the beginning of the stage.

At the end of the stage, the students were evaluated by each manager and the worst evaluated half of the team would be reallocated to another team in the next stage. The purpose of this was to simulate what happens in companies. Periodically, employees are evaluated and given feedback to understand their strengths and weaknesses and work on them. In addition, these evaluations show each employee's performance and what their individual skills are. Based on this information they are reallocated to other teams as the company needs.

Stage 2: Planning and Modeling

In the beginning of stage 2, the teams were incomplete, so another dynamic of "hiring" students was conducted. Each manager and the remainder of their teams interviewed the unallocated students, both managers and unallocated students established a list of priorities based on these interviews. Finally, crossing the information of both lists, the students were assigned to their new team.

This stage of the project aimed at elaborating a project plan to support the later stages and also model the system using UML based on the defined requirements. Similarly to stage 1, the teams were free to contact the stakeholders at any time to ask for questions. The deliverable artifact expected to be handed in 4 weeks later was the project plan and the design document, containing all UML models.

In the end of this stage, the teams had undergone changes of students again. Those who had stayed in the previous stage should be relocated to a new team.

Stage 3: Prototyping and Systems integration

Stage 3 also began with a "hiring" dynamic. One more time each manager and the remainder of their teams interviewed the unallocated students, and established a list of priorities based on these interviews. The students were assigned to their new team, after crossing the information of both lists.

This stage of the project concerned the development of a functional prototype and the integration of all systems. The teams had to communicate with the other teams of the same group to reach an agreement on how to integrate those systems considering graphic user interface and also the design artifacts, since it is important to integrate those systems visually and functionally.

After 4 weeks, the students should hand in the following artifacts: functional prototype and UML artifacts integrating the five systems. The teams did not change at the end of this stage.

Stage 4: Presentation to stakeholders

The fourth stage was the quickest one and lasted only two days. The main idea was to present how the project was conducted and the functional prototype to stakeholders to receive feedback from them.

4 EVALUATION

Aiming to evaluate the applied approach, we gathered information from both graduate and undergraduate students. In this section, we present a summary of all this information. Firstly, we present the main feedback from undergraduate students regarding the approach applied in the SE course, and then, the feedback from graduate students, who participated in the approach as project managers, guiding the undergraduate students.

4.1 Undergraduate students

To understand the undergraduate students perception on the use of the approach in learning Software Engineering, we conducted a survey composed of eight mandatory Likert scale questions and one optional open question (Table 1). The questionnaire was sent by e-mail to all students who were managed by the authors of this paper. As results, we obtained and analyzed answers from 42 students[1]. The results are summarized and presented in Figure 1.

[1]The data is available at icmc.usp.br/e/ebfbe

Concerning the motivation to learn SE, most students (56.0%) agreed this learning approach has motivated them. On the other hand, we had a considerable number of students (28.6%) who neither agreed nor disagreed. 14% of students declared the approach did not motivate them. When we compare this first question with students' comments, we noticed that some students did not understand the real motivation to simulate a company inside the classroom. Other students complained the project required much time beyond what was planned for the course.

Table 1: Survey Questions

Question	Type
(1) Did you feel motivated to learn Software Engineering with a PBL approach?	Likert scale (Mandatory)
(2) Did you consider the project manager important during the project conduction?	Likert scale (Mandatory)
(3) Did you feel confident in the project manager during the project conduction?	Likert scale (Mandatory)
(4) Did you believe that the use of the PBL approach influenced your learning process?	Likert scale (Mandatory)
(5) Did the real-world project experience motivate you to work in a software development company?	Likert scale (Mandatory)
(6) Was the contact with real customers important?	Likert scale (Mandatory)
(7) Was the work with real-world project important?	Likert scale (Mandatory)
(8) Did you feel satisfied with the final product delivered in the Software Engineering course?	Likert scale (Mandatory)
(9) Do you have additional comments, criticisms, compliments or suggestions regarding the approach used in the Software Engineering course?	Open (Not mandatory)

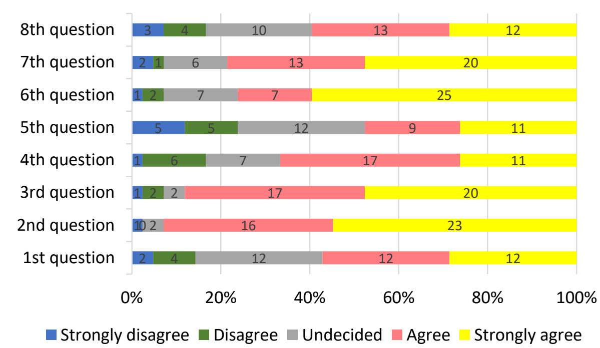

Figure 1: Results of the eight likert scale questions

Regarding the importance of the project manager, results indicate 54.8% of students totally agreed and 38.1% partially agreed. In general, they considered the manager presence important to lead their projects. When asked if they felt confident about manager leadership, more than 87.0% declared they have enjoyed managers' participation during the projects.

With respect to effectiveness of their learning process using the PBL approach, results indicate more than 40.5% of students partially agreed on this statement and 26.2% totally agreed. 16.7% neither agree nor disagree and only 16.7% of students disagreed. Considering students' comments, they reported a very good experience with the approach and, they also declared other courses should adopt it. In addition, some students emphasized the approach made the lesson less expository, which automatically shifted the responsibility of the learning process from the teacher to the students.

We also found out not all students agreed whether the experience in software development motivates them to pursue a career. Results indicate that most students (28.6%) were not sure on this statement, but a representative amount agreed with it, either partially (21.4%) or totally (26.2%). In addition, 23.8% of students did not feel motivated and disagreed, both partially (11.9%) and totally (11.9%). When asked about the importance of the relationship they established with clients, most of students agreed it was essential, 59.5% totally agreed and 16.7% partially agreed. On the other hand, 4.8% totally disagree, 2.4% partially disagree and 16.7% were not sure about it. In both questions, it was not possible to relate the results with the comments provided by the students.

When asked whether they felt that working with a real project was important, a large number of students agreed, either partially (31%) or totally (47.6%). Some of them were not sure (14.3%) and few disagreed totally (4.8%) or partially (2.4%). Relating to the comments, the students highlighted the experience was positive but also complained that the time required to complete the project required more hours of dedication than that requested officially in the course.

Finally, we asked students if they felt satisfied with the project developed in the course. More than half of the students totally (26.8%) or partially (31%) agreed with this statement, allowing us to conclude that even with all the challenges, such as time, different profiles, managers, abilities and expectations, it was a motivating experience and students enjoyed the project. In addition, some students disagreed totally (7.1%) and partially (9.5%), and some were not sure (23.8%). As comments, students mentioned that it would be interesting if the product could be implemented and they receive feedback.

Aiming to verify the internal consistency of the survey, after its application, the *Cronbach's Alpha Test* was used [8]. The value for the applied questionnaire, applying the test, was 0.90 [2], which according to Streiner [16] indicates that the questionnaire has high reliability in the space that it was applied.

We based the analysis of our last question on some procedures of Grounded Theory [15] to analyze students' comments based on the concept of coding, such as: *open coding* makes possible identification of concepts that are separated into discrete parts for analysis; and *axial coding* handles connections among codes and groups them according to their similarities.

Our data sample consisted of students' comments on their experience in using PBL approach and designing a software project. The general idea with this project is to give students a snapshot from the real world to design a software to real clients. We asked an open question and students were encouraged to describe their experiences and suggestions to improve the approach.

All answers were organized in a single file and each sentence was analyzed to derive the codes by using open coding procedures. With respect to PBL approach, some students declared the following: *"The approach was different from what I have seen in my graduation classes. In general, it was really good"*, *"The idea was really good. Learning theory classes in a practical way is much more motivating than traditional classes in which we do not know why we are learning the subject"*, and *"I believe practical approach is very good because we can learn concepts not just in theory"*. These pieces of sentences were clustered in the code `Practical approach motivates learning`.

In addition, students described their negative experience with the real world project, such as: *"As a consequence of the poor distribution/division of the work, the time that each student dedicated to develop the project was completely disproportionate"*. These sentences were grouped into `Activities poorly organized`. Students also complained about the projects' dynamics and the time needed, as follows: *"One point to be settled was the constant change of group members. This affected to final outcome of the project. It was also discouraging"*, *"I believe that the workload should be revised since much time has been spent to build the project"*. These sentences were coded in `Unmotivating group dynamics` and `Time needed`. Morever, another student said *"We are not trainees, we are students. It makes no sense to try to simulate a company in a university"*, which were related to `Students are not trainees`.

Students also reported their experience with the project manager, such as *"Some managers did not lead"*, which was coded into `Manager leadership`. They also described their experience about the classes: *"There was a lack of teacher contact with the students for motivation and teaching"*. These information were clustered into `Lack of theory classes`. Furthermore, students also reported their exposure with real clients by detailing *"Working together with a real client was a very positive experience"*, that was coded into `Contact with real-world clients`. One student also declared *"The only thing that would make me satisfied is the final implementation of this work. I hope stakeholders get the final product done"* that was coded in `Client's project Implementation`.

In summary, we identified 10 codes and using axial coding procedures, we aggregated each code into categories based on their similarity. We performed open and axial coding several times aiming to refine the emerging codes and categories. Furthermore, we mitigated an eventual bias in the coding process by discussing the codes and categories among researchers until they came to an agreement for all the concepts found. The categories are presented next, followed by the assigned codes.

- PBL: Motivates learning
- Lectures: Lack of theory classes
- Clients: Client's project, Contact with real-world clients
- Manage: Manager leadership
- Project: Activities poorly organized, Unmotivating dynamics, Students are not trainees, Project poorly organized, Time needed.

The categories and codes emerged through Grounded Theory procedures allowed us to suggest some assumptions from our findings. For example, there was an overall agreement by students that PBL approach motivates the learning by exposing students to real-world problems. Students were satisfied to deal with real clients and this motivated them to deliver a good final project. However, students expressly referred to the poorly organization of the whole software project including: group dynamics simulating hiring and firing employees; project activities, such as interview with client, requirement elicitation, UML software design, and prototyping; and the time needed for performing all the tasks. Based on such data, students were more likely to design a success software project when they were not forced to change group. This is true regardless the short time needed to learn the whole new project. However, we

[2]The data used to calculate alpha is available at icmc.usp.br/e/88406

can also infer students felt uncomfortable when they were "fired" discouraging them to work with the new group.

4.2 Graduate students

We also asked the graduate students their feedback about the applied PBL approach. Each student briefly described his/her observations as follows. We identify the students with the letters "GS" followed by a random number.

GS1 stated *"I had a very positive experience as a project manager in the SE course. Most of the students were dedicated and they came up with great ideas. Disagreements were discussed among the members and all activities were shared. In addition, we had a great time with our clients who were always available to give more details about the projects. In the end, both groups came up with great projects and they got compliments from the clients. On the other hand, I had to support and review all students activities which took me a lot of time including weekends. I think all the activities proposed must be rethought aiming to balance the work for all the students."*. In a related perspective, GS2 said *"The experience I had managing and working with students was one of the most enriching of my academic background so far. The students who were part of my group were very dedicated to the course, often asking good questions about the content and concerned about developing a project with quality. We spent some extra classroom time together to do the activities as we wanted, but I believe the effort paid off. Also, I really enjoyed the real project, I believe it was a great motivating factor to accomplish the tasks. My complaints are about the change of groups, because our work was progressing well, and in my opinion, had the changes not occurred we would have been able to deliver a much more complete project."*.

GS3 believes that *"Simulating a project from a real context within the SE course was an enriching experience. I could see that the students were interested and were instigated to put into practice the knowledge acquired in class. They were concerned about the product that was being developed and I believe that this is due to the fact of the real application it had. As a limitation of the lived experience, I point out the fact that the course has been offered in parallel with others, as a standard in universities. So the students did not have full time to devote themselves even more to the project, as is done in industry, for example."*. On the other hand, GS4 states that *"Overall, the experience I had in this course was positive. Through the approach applied, I could manage software projects and guide undergraduate students in performing project activities. Not only was I able to share my knowledge with my students, but I also learned from them. I believe this kind of experience is very important because it breaks the hierarchical aspect between undergraduate and graduate students. Furthermore, I also believe students had an unique opportunity during their course, which will definitely help them when they leave the university to go to industry. Although the great experience, I have faced some challenges since I had to play a role that goes beyond the manager activities, writing reports, redoing diagrams, requiring attention because of the delayed activities and changing the deadlines, and so on."*. Finally, GS5 said *"I believe the idea of simulating a real-world environment is really important and interesting. I had some issues dealing with students who were not fully committed to the course. Considering my experience in a company environment, when a employee is not committed and the manager can not solve the problems, s/he can escalate the problem to a higher management, but it didn't occur in this course and the manager ended up doing the activities when the responsible did not. Also, the managers were responsible for activities that in a real environment they wouldn't be such as content review or even teaching how to design some UML diagrams. The roles of manager and course tutor were totally mixed up. Although the idea is interesting, the proposal needs to be refined so that in practice each person fulfills her/his role as it is in a real environment."*.

We can notice graduate students highlighted both positive and negative aspects. We can point out some positive aspects, such as: use of real-world projects, contact with stakeholders and business environment simulation. Regarding the negative aspects, we can mention: difficulties managing undergraduate students, exhaustive review tasks and graduates had to teach some software engineering and UML modeling concepts to undergraduates, which they believe is the role of a teaching assistant.

5 CHALLENGES AND LESSONS LEARNED

In this section, we present the challenges we identified to apply this approach and what we learned as managers of undergraduate students. The challenges we can highlight are:

- **Student's profile:** Manage a team composed of 8 to 10 students was one of the greatest challenges we identified due to the difference in their profiles, i.e., each one had a different interest in the course, some of them wanted to learn while others only wanted to complete a mandatory course. Besides that, the students' prior knowledge also impacted the project, since some had a greater ability to develop activities, such as requirements elicitation and diagrams development, while others had a very basic vision. In general, we consider as a challenge not their level of knowledge, because leveling could be performed, but rather the great difference among them, unbalancing the team's progress.
- **Activities management:** Although the three first stages of project had similar time to be developed, the amount and complexity of artifacts that were supposed to be developed were different for each one, leading some to be prepared with higher quality and others to be completed more quickly, since there were dependencies between them and it could not take a longer period. Moreover, due to the difference in the workload, it was difficult to assign individual tasks for students in all the stages because they still had projects from other courses.
- **Student's maturity:** Another big challenge was the undergraduate students' lack of maturity, which often failed to manage their own tasks, overloading the manager asking for help and assistance. In this sense, the challenge of managers was the amount of time assigned to manage the team, besides assigning clear and objective tasks, so that the students were motivated to solve them by themselves.
- **Planning and changes:** The lack of planning in the course was also a challenge, both for undergraduates and graduates, because the activities that should be carried out were not always well defined. For this reason, managers had uncertainties about assigning activities correctly among the students so that they would perform quality work without wasting too much time. This lack of organization also discouraged them because they did not see the manager's steady presence in leadership.

Furthermore, the change in the group members with the dynamics of layoffs also proved to be a challenge, since even with the rationale that in a company employees are regularly relocated to other projects, the students were still not motivated to change groups, because the time needed to adapt to a new project to perform activities with quality was not compatible with the time allocated to complete the stage. Finally, the lack of communication among the managers caused the groups to be in different stages and, therefore, when the students were transferred to another team they were sometimes faced with delayed teams.

• **Integration:** Due to the difference in teams progress, another challenge was to integrate all projects, since each group had a significant number of students, which had different opinions and had made several decisions to complete their tasks.

As presented, several challenges were faced with the objective of delivering quality projects and, at the same time, guarantee students' learning of the main SE concepts. Despite considered challenges, they were essential to build a set of lessons learned that could improve the approach.

Among the lessons learned, we can highlight the planning of the approach as the primary activity for its success. As presented in the assessment results by students, the approach improves motivation, learning support, and the presence of the manager makes the students feel more confident. However, to maximize all of these positive aspects, good planning is essential when applying the approach, define in advance exactly what the delivery artifacts are and how much they should be delivered, plan the delivery according to the amount of available students, balance workloads, and distribute students into groups according to their interests. In addition, the exchange of groups is also an interesting aspect, but it should occur less often and less abruptly so that students could remain more motivated to work on the projects. Another option would be to keep the same groups during the whole course.

The proximity of managers to students is also a positive point to be highlighted from our approach. Groups in which managers were present most of the time, even if overwhelmed, had an interesting income and were able to better discuss the activities that should be developed. The role of manager who only assigns tasks and returns to receive them does not fit well in this approach because it breaks an important goal that is to bring students closer to the contents that need to be learned. In this sense, graduate students have an important role, since they are SE researchers and have mastery of the content taught.

Besides that, we also point out the use of projects dealing with real-world problems that can be analyzed by students with data, forms, and interviews with clients can be a positive impact factor to better engage them in learning, since the students see the usefulness of what they are developing, unlike other developed undergraduation projects that focus only on learning but ignore the application.

6 CONCLUSIONS AND FUTURE WORK

This paper presented a PBL approach for a Software Engineering course, discussing its use as a supporting mechanism for teaching and learning. The novelty of the approach was the combination of PBL with project management.

The inclusion of a real context project within the classroom showed that PBL has the ability to bring innovation and dynamism in teaching software engineering. Unlike the classes that bring the content only in an expository way, the approach used instigates the students to assume a role of pro-activity and responsibility in what concerns the acquisition of knowledge. In general, students were enthusiastic and positive about the PBL approach and the importance of using real problems. Although the approach still requires improvements, the applied version has fulfilled requirements for teaching Software Engineering.

We expect this approach brings students closer to the real context of software development and, consequently, that students better understand the topics of the course and more important, feel more motivated to study it.

As future work, we aim at refining the approach to mitigate the faced challenges such as: unbalanced groups of undergrad students, different times of each stage, lack of planning, integrating the projects. We also intend to conduct a more complete evaluation to measure the approach effectiveness regarding the learning process. We intend to perform a experimental study in order to compare our approach with a PBL approach to teach SE without the participation of a project manager. In this experimental study the idea is to analyze several aspects such as motivation, learning performance, and so on.

ACKNOWLEDGEMENTS

The authors would like to thank Procad-CAPES (Grant 71/2013), FAPESP and CNPq. Our greatest thank and gratitude goes to the undergraduate students who contributed to this project.

REFERENCES

[1] ACM/IEEE-CS. 2013. Computer Science Curricula. www.acm.org/education/CS2013-final-report.pdf. (2013).
[2] ACM/IEEE-CS. 2014. Software Engineering Curricula. http://www.acm.org/binaries/content/assets/education/se2014.pdf. (2014).
[3] ACM/IEEE-CS. 2016. Computer Engineering Curricula. http://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf. (2016).
[4] A. Baker, E. O. Navarro, and A. Van Der Hoek. 2005. An experimental card game for teaching software engineering processes. *Journal of Systems and Software* 75, 1 (2005), 3–16.
[5] C. C. Bonwell and J. A. Eison. 1991. *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports.* ERIC.
[6] J. Chen, H. Lu, L. An, and Y. Zhou. 2009. Exploring teaching methods in software engineering education. In *Computer Science Education ICCSE.* IEEE, 1733–1738.
[7] K. Claypool and M. Claypool. 2005. Teaching software engineering through game design. In *ACM SIGCSE Bulletin*, Vol. 37. ACM, 123–127.
[8] L. J. Cronbach. 1951. Coefficient alpha and the internal structure of tests. *Psychometrika* 16, 3 (1951), 297–334.
[9] IEEE. 2010. *ISO/IEC/IEEE 24765: 2010(E): Systems and software engineering – Vocabulary.* IEEE.
[10] H. Kerzner. 2013. *Project management: a systems approach to planning, scheduling, and controlling.* John Wiley & Sons.
[11] R. Kessler and N. Dykman. 2007. Integrating traditional and agile processes in the classroom. In *ACM SIGCSE Bulletin*, Vol. 39. ACM, 312–316.
[12] R. Matthews, H. S. Hin, and K. A. Choo. 2012. Merits and Pitfalls of Programming Learning Objects: A Pilot Study. In *10th International Conference on Advances in Mobile Computing Multimedia (MoMM).* ACM, Bali, Indonesia, 293–296.
[13] A. Munns and B. Bjeirmi. 1996. The role of project management in achieving project success. *Inter. Journal of Project Management* 14, 2 (1996), 81 – 87.
[14] R. Riding and S. Rayner. 2013. *Cognitive styles and learning strategies: Understanding style differences in learning and behavior.* Routledge.
[15] A. L. Strauss and J.t M. Corbin. 1998. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory* (2 ed.). SAGE.
[16] D. L. Streiner. 2003. Being Inconsistent About Consistency: When Coefficient Alpha Does and Doesn't Matter. *J. of Personality Assessment* 80, 3 (2003), 217–222.