The Use of Industry Training Strategies in a Software Engineering Course: An Experience Report

Carlos Portela, Alexandre Vasconcelos
Informatics Center (CIn)
Federal University of Pernambuco
csp3@cin.ufpe.br, amlv@cin.ufpe.br

Sandro Oliveira
Post-graduate Program of Computer Science (PPGCC)
Federal University of Pará
srbo@ufpa.br

Maurício Souza
Department of Computer Science (DCC)
Federal University of Minas Gerais
mrasouza@dcc.ufmg.br

***Abstract*—The software industry presents dissatisfaction in relation to the level of recently graduated professionals. Consequently, software companies end up investing in the training of these professionals in order to develop skills related to the software development process. This paper presents an experience report of an adaptation of industry training practices in a course of Software Engineering (SE). Initially, we mapped SE topics of the ACM/IEEE curriculum guidelines to the specific practices of the CMMI-DEV model. Then, we carried out a survey of training strategies with 10 consultants in Software Process Improvement (SPI), who also act as SE professors. Finally, these strategies have been adapted and incorporated in a SE Course. We compared the impact of these learning strategies in relation to traditional teaching approaches, regarding the student perception of learning. We observed an increase of approximately 20 percent in student perception of learning in relation to the SE topics with industry training strategies.**

Keywords—industry training strategies; software engineering teaching; software engineering learning

I. INTRODUCTION

Software Engineering (SE) is a discipline that is concerned with the application of theory, knowledge and practice to the effective and efficient development of software systems that meet the user requirements [1]. The curriculum guidelines of the ACM/IEEE [1] and Brazilian Computer Society (SBC) [2] recommend that the undergraduate courses should include the topics of SE knowledge units that allow the development of the competences expected for professionals in this area. For example, in order to develop students' ability to work in groups, we can apply the topics of the Project Management unit, and to develop written and verbal communication skills, we can use techniques related to the Requirements Engineering unit.

In addition, these curriculum guidelines emphasize that the professional competences emerge through the theoretical study of these units and the practical application of their concepts. It is critical that students understand the relationship between theory and practice. Moreover, they must understand how the theory influences the practice and vice versa. However, in academia the teachers tend to be very theoretical [3]. This implies that academia often neglects the development of skills required by the industry such as communication, leadership, conflict resolution, and group dynamics [4].

In this scenario, the software industry is dissatisfied with the level of preparation of the recently graduated professionals [5]. Therefore, software companies invest in training and certifications of these new professionals to repass the specific practices of Software Engineering areas [4]. According to Meira [6], companies train their professionals in the rhythm of the changes of theoretical and practical platforms of the business, through formal training in the beginning of the career and other hundreds of explicit learning during the execution of their activities. In addition, there are many opportunities for implicit learning.

Gimenes [7] emphasizes that we are facing an educational context that drastically questions the teaching/learning forms. Regarding teaching strategies, ACM/IEEE [1] and SBC [2] emphasize the need to go beyond the expository classroom format, since it does not favor the effective student learning. In this sense, it is important to consider the variation of teaching and learning techniques [1]. Seeking to change the current dynamics of teaching and to meet the need for professional training in the industry, this paper proposes adapting the training and evaluation practices of the industry to the academic context.

Since industry complains that undergraduate courses do not properly train professionals and adopt their own strategies, a possible solution would be to anticipate the application of these training practices, considered effective by the industry, in the training of undergraduate students [6]. To this end, we identified which training strategies are used by Software Process Improvement (SPI) consultants to develop competences and abilities in professionals. Given the various limitations of the academic environment, we intended to adapt these practices according to the resources available in this environment.

The goal of this paper is to report and discuss the results obtained in the application of industry training strategies in a Software Engineering course. In this study, we expect to contribute to an attempt of reducing the gap between the teaching approaches in academia and the development of skills expected in the industry. In order to meet this demand, both to identify industry practices and to identify training strategies, we carried out a survey with SPI consultants who also act as SE professors. Therefore, this research intends to contribute to the identification of training practices that enable students to

2377-570X/17 $31.00 © 2017 IEEE
DOI 10.1109/CSEET.2017.16

develop these professional competencies, in order to reduce the shortage in the training of professionals and, consequently, to meet the demands of the software industry.

In addition to this introductory section, Section II presents the research methodology, highlighting the topics of SE suggested by the curriculum guidelines, a mapping between these topics and the specific practices of quality models, and the survey about training strategies. Section III reports the application of the training strategies of the industry in a SE course. The analysis and discussions about the results obtained in this research are presented in Section IV, in addition to the main lessons learned. Section V presents the main threats to validity. Finally, Section VI presents the contributions, conclusions and future works of this research.

II. Research Methodology

According to Meira [6], it is necessary to review what and how to create opportunities to learn SE. Thus, we carried out a survey that sought to identify the most adopted topics of the curriculum guidelines of ACM/IEEE [1] and SBC [2], in order to refine the amount of content and, consequently, to use more effectively the time available for the Software Engineering course [8]. The Subsection II-A describes the topics covered in this research. Besides that, the Subsection II-B describes the mapping between these topics and practices of quality models and the Subsection II-C describes the survey with consultants in SPI.

A. *Identification of SE Topics*

According to SBC [2], the topics related to Software Engineering education are: Software Development Process; Software Development Lifecycle; Software Quality; Techniques of Planning and Management of Software; Software Configuration Management; Requirements Engineering; Methods of Software Analysis and Design; Software Quality Assurance; Verification, Validation and Testing; Maintenance; Documentation; Development Standards; Reuse; Reverse Engineering; Reengineering; Software Development Environments.

According to the ACM/IEEE [1], the knowledge units related to Software Engineering education are: Software Process; Software Project Management; Tools and Environments; Requirements Engineering; Software Project; Software Development; Verification and Validation; Evolution of Software; Reliability of Software; Formal Methods. Each of these units is composed of a set of Topics related to the expected learning.

In this work, we chose to adopt the ACM/IEEE classification for several reasons. Firstly, the ACM/IEEE curriculum guidelines [1] is recognized and adopted internationally. Additionally, the current version of these guidelines are more recent than the SBC curriculum [2]. In addition, due to the large number of subjects in Software Engineering education and the low availability of teaching hours, it is necessary to prioritize which topics will be addressed. It is necessary due to the profile of the professional that one wishes to train [9]. In this context, the ACM/IEEE curriculum recommends 10 knowledge units, while the SBC curriculum recommends 16 topics. Finally, it is noticed that all of the SBC curriculum topics are direct or indirectly covered by the ACM/IEEE curriculum.

Analyzing the ACM/IEEE curriculum [1], we identified 83 topics and 125 expected learning to the SE area, classified and organized in 10 knowledge units. We conducted a survey [8] with students and professors regarding the adoption and learning of SE topics. The purpose of this survey was to refine the number of topics. Thus, we received responses from 70 participants, being 23 professors and 47 students. Participants represent 12 states in Brazil, with 50% of institutions in the Northeast, 25% in the North, 15% in the South, 5% in the Midwest and 5% in the Southeast. Most of the participants, 80%, are from public institutions and 20% from private institutions.

From the data analysis of this survey, we identified the 6 units most adopted by the professors in the scope of the SE course. Then, we correlate the percentage of topics adopted with the percentage of students' learning, as shown in Fig. 1.

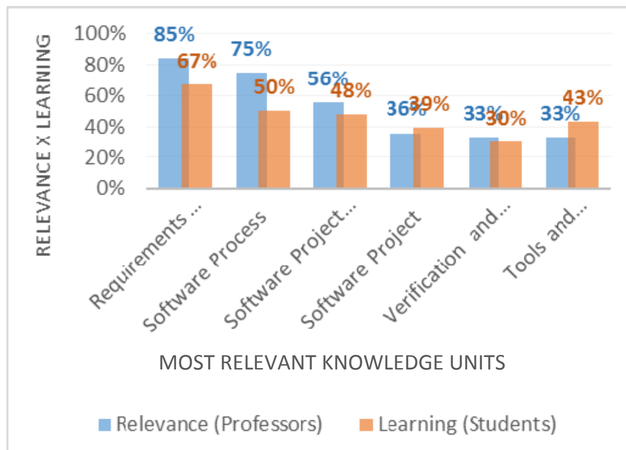

Fig. 1. Correlation between Relevant Topics and Learning

According to the results of the survey, we observed that Requirements Engineering is the most relevant unit, since it is widely considered in Software Engineering curricula by 85% of the professors, and effectively learned by 67% of the interviewed students. Next, we highlighted the Software Processes units, taught by 75% of the professors and learned by 50% of the students, and Software Project Management, taught by 56% of the professors and learned by 48% of the interviewed students.

If the teacher has a limited time and it is necessary to prioritize some unit, we suggest these 6 knowledge units. Therefore, we defined them as the focus of application of this research, being related in the survey of training strategies of the industry with consultants.

B. *Maping Between Topics and Practices*

This mapping aimed to relate the SE topics with the best practices adopted by the industry in the software process development. Quality models such as the Capability Maturity Model Integration for Development (CMMI-DEV) and the

Brazilian Reference Model of the Software Process Improvement for Software (MR-MPS-SW) are based on area standards and present, respectively, a set of process areas/processes and specific practices/expected results that serve as a reference for software companies.

In order to map these curriculum guidelines and quality models, the curriculum of the ACM/IEEE [1] which, as described in Subsection II-A, covers the Topics of the SBC [2] was used as a basis. Similarly, it was decided to consider only the CMMI-DEV [10] model for this mapping, as it served as reference and has adherence to the MR-MPS-SW [11].

The first stage of this mapping occurred at the level of structure and concepts, as described in Table 1.

TABLE I. MAPPING BETWEEN CMMI-DEV AND ACM/IEEE

CMMI-DEV Concepts	***ACM/IEEE Concepts***	***Description***
Process Area	Knowledge Unit	A process area is a set of related practices and a knowledge unit is a set of related topics.
Specific Goal	Not identified	There is no set goal for a set of topics.
Specific Practice	Topics	A specific practice is the description of an activity and a topic is the description of a knowledge.
Subpractices	Learning Outcomes	A subpractice is a detailed description of a specific practice and learning outcomes are related to a specific topic.

A process area is a set of related practices that, when implemented collectively, satisfy a set of goals considered important for the improvement of an area, such as Requirements Management (REQM). Similarly, a knowledge unit is a set of related topics that makes up the knowledge of a SE area, such as Requirements Engineering.

A specific practice is a description of an activity that is considered important for meeting a specific associated goal, such as "SP 1.1 Understand Requirements". On the other hand, a topic is the description of a knowledge related to an unit, such as "Elicitation of Software Requirements".

Finally, a subpractice is a detailed description that provides guidance for interpretation and implementation of a specific practice, such as "Analyze requirements to ensure that they are necessary and sufficient". Expected learning is related to the outcome of teaching a specific topic, such as "Conduct a review of a set of software requirements to determine the quality of the requirements with respect to the characteristics of good requirements".

C. Survey of Training Strategies

In order to identify industry practices and training strategies, we conducted a survey with 10 Software Process Improvement (SPI) consultants. The purpose of this survey, available in Google Forms (*https://goo.gl/FSfsiQ*), was to identify, in the perspective of specialists in the area (SPI consultants and quality models implementers, who also act as undergraduate professors), which the capacity and the evaluation strategies are adopted in SPI programs. Table 2 shows the questions asked to the consultants in this survey. Questions 2 and 3 were asked for each knowledge unit, according to their correlation with process areas of the quality models.

TABLE II. QUESTIONS FOR THE SURVEY OF TRAINING STRATEGIES

Question	***Identified Responses***
1. Which models of software process improvement you implement?	( ) MR-MPS-SW ( ) CMMI-DEV ( ) Others
2. Which is the training strategy adopted for the processes related to Software Engineering?	( ) Workshop ( ) Group dynamic ( ) Mentoring ( ) Coaching ( ) Others
3. How do you evaluate the learning in these strategies?	( ) Objective Test ( ) Subjective Test ( ) Participation ( ) Frequency ( ) Others

For the Question 2, we considered the following definitions:

- Workshop: a seminar or an intensive course, of short duration, in which techniques and skills are demonstrated and applied;
- Group dynamics: an instrument that is part of a training process, which enables the creation or re-creation of knowledge;
- Mentoring: when a more experienced professional guides and shares experiences and knowledge with less experienced professionals;
- Coaching: consists in the development of competences and abilities from training in specific techniques.

In order to delimit the scope of this survey, the consultants answered only about the knowledge units identified as the most relevant in the survey conducted by Portela et al. [8]. The training practices listed in the responses were identified in the works of Gnatz et al. [12] and Garg and Varma [13] and the training strategies were based on the work of Prikladnicki et al. [3].

From the analysis of the answers, the most adopted practices were identified, as presented in Fig. 2.

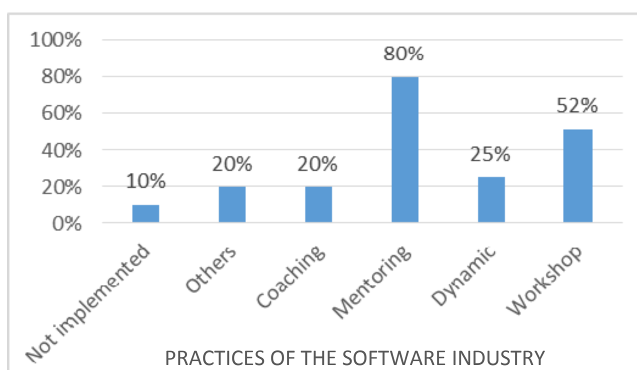

Fig. 2. Percentage of Training Practices Adoption

The most adopted training practice was mentoring, which consists of the consultant to orient and share their experiences and knowledge in the SPI processes with the professionals of the target company of the improvement. The second most adopted practice was workshop, where the consultants execute a short seminar, presenting techniques and skills, and demonstrating how to apply them.

From the identification and analysis of these practices, we discussed the strategies of their adaptation to teaching in a SE course. It was also adopted the continuous representation of CMMI-DEV, that defines which process areas will be the focus of improvement. Thus, students and professors will be able to delimit the scope of their efforts, reducing the amount of SE topics to work on the development of SE competencies and abilities.

After listing the training practices adopted in the process implementation, the consultants listed the evaluation strategies used in this process. The strategies identified are presented in the Fig. 3.

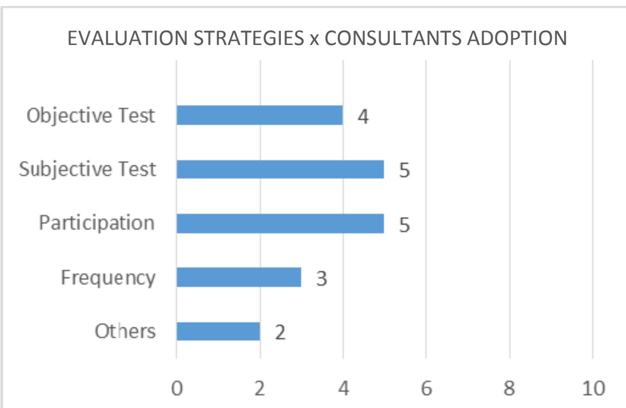

Fig. 3. Evaluation Strategies adopted by Consultants

We observed that the consultants mainly adopt subjective tests and the student participation in classroom to evaluate the learning. In addition, they perform objective tests and consider frequency (attendance) in order to assess impairment. The consultants also cited that they evaluated the professionals by observing the accomplishment of the practical activities in a project. We also adopted these assessment practices in the SE course reported in this paper.

III. EXPERIENCE REPORT

In this section, we describe the experience of adapting industry training practices in a SE course in an undergraduate program in Computer Science. In order to plan and execute this experiment, we followed five steps, as shown in Fig. 4.

Initially, in Step I – Study Design, the research planning was carried out in order to identify the involved variables, the research questions and the methods of data analysis. In Step II - Environment Preparation the training of the teachers in the practices of the industry was realized. In addition, the instruments for data collection and analysis were prepared at this stage.

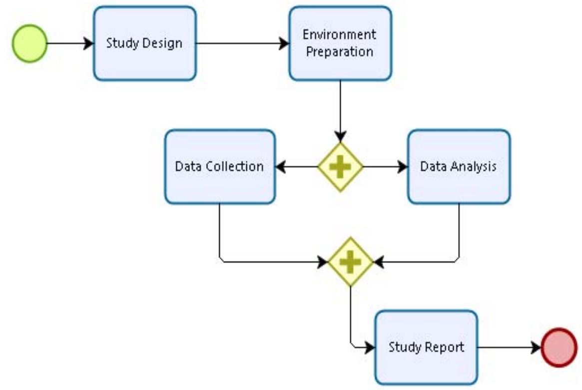

Fig. 4. Steps of the Study

In the Step III - Data Collection was carried out the observation and collection of relevant data through the established instruments. Concurrently, in Step IV - Data Analysis the collected data was analyzed in order to describe the phenomenon observed. Finally, in Step V - Study Report, a research report was generated, presenting quantitative and qualitative data that was discussed and analyzed by the researchers, teachers and students involved in the experiment.

A. *Research Goal and Variables*

In the undergraduate program in Computer Science of the University Center of the State of Pará (CESUPA), the Software Engineering area is mainly addressed in 3 courses: Software Engineering I (SE-I), Software Engineering II (SE-II) and Integrated Project in Software Engineering (IPSE). The focus of this report is the IPSE, with 60 classroom hours, whose learning objective consists of presenting the topics of the SE knowledge units, integrating theory and practice from a development project. Thus, students assume a role (analyst, programmer or manager), plan, codify and deliver an application.

Since this course is the context of the application of the training strategies of the industry, we defined the following Research Goal (RG):

- RG: *Does the use of industry training strategies allow an increase in the learning of topics in the Software Engineering area?*

In order to meet this RG, the following variables were selected: (i) the training strategies; (ii) the students; (iii) the learning; (iv) the topics. Related to the variable (i), we approached the training strategies described in Subsection II-C. For the variable (ii), 14 students from the 6th semester (a whole class), 2 veteran students (8th semester), 1 professor and 1 researcher/observer were involved in the study. These students had already finished the SE-I course, which aims to introduce the SE basic concepts. Besides that, they were coursing in the same period the SE-II course, which aims to present lifecycle models and enable the students to define development process from pre-defined models.

Regard to variable (iii), we chose to measure the student learning at the pre-course and post-course moments in order to compare the learning increase. Additionally, we measured the learning of the veteran students who attended the course in the second half of 2016, trying to compare the results obtained in this experiment with a traditional teaching approach.

Finally, related to variable (iv), the course syllabus of the IPSE addressed the following knowledge units: I. Requirements Engineering; II. Software Process; III. Software Project Management; IV. Software Projects; V. Software Verification and Validation; VI. Tools and Environments. In order to a better report of the experience of using the industry training strategies, this paper will focus on the topics of the Software Process knowledge unit. However, the results will be described considering the topics of these 6 knowledge units.

B. Preparation of the Enviroment

Before starting the course, the training and evaluation practices were presented to the responsible professor, who is a PhD student in SE and has been working for more than 10 years in the SE area. He also carried out the reading of related works and made the necessary adjustments for the application of the practices in the teaching plan of the IPSE course.

In the first day of class, the students were divided into 2 teams (A and B) of 7 students, where each team had 1 specific client/project and 1 veteran student to perform coaching. The professor assumed the responsibility of mentoring the two teams, being responsible for identifying the group dynamics related to knowledge units and inviting professionals to give workshops on support tools.

The project of the Team A consisted in the development of a mobile application to access CESUPA's online student system, whose clients were the students of the institution. The Team B project consisted in the development of a web application to register ideas of projects and professionals interested in integrating the team of these projects, whose client was the professor of the SE-II course.

C. Data Collection

Initially, these students completed a questionnaire about the prior knowledge (before beginning the course) of Software Process topics. For each topic of this unit, we asked what the degree of student knowledge on a Likert scale from 0 to 5:

- 0 - I know absolutely nothing;
- 1 - I know vaguely;
- 2 - I know the basics;
- 3 - I know moderately;
- 4 - I know a lot;
- 5 - I know in depth.

In addition, the students created conceptual maps [14] and data dictionaries of SE knowledge units based on their prior knowledge. Figs. 5 and 6 present, respectively, a conceptual map and data dictionary of the Software Process unit developed by a student at the beginning of the course (before applying the training practices).

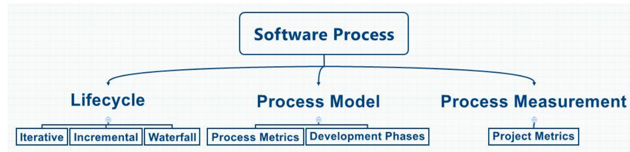

Fig. 5. Example of Conceptual Map (Pre-Course)

Cascade: *Model of software production.*

Lifecycle: *Period in which the project is receiving maintenance.*

Development Phases: *Steps in which a development team is producing software.*

Incremental: *Model of software production.*

Iterative: *Model of software production.*

Process Measurement: *Evaluate the performance of a given process.*

Process Metrics: *I do not know how to respond.*

Process Models: *I do not know how to respond.*

Fig. 6. Example of Data Dictionary (Pre-Course)

The topics of the Software Process knowledge unit were taught during 3 weeks, totaling 9 classes (9 hours or 15% of the total of course hours). In the class #1, the study of this unit began with the identification of a problem related to the project to be developed in the course, such as "defining a process for the development of a mobile application that accesses the online student portal". In class #2, the students carried out readings of papers on process definition and in class #3 they attended a workshop, about the Scrum agile framework, given by a project manager who works in the software industry.

In classes #4 and #5 the students were able to experience the practical application of these concepts from the aircraft factory activity [3], which consists of producing paper airplanes, where the production process is the team's decision. This activity was conducted by the professor himself. In class #6, students registered expectations regarding the application of the concepts learned in the practical project of the course.

Finally, in classes #7 and #8, students integrated the skills acquired in these stages in the software project of the SE course, modeling a process from the notations of the Software Process Engineering Metamodel (SPEM) standard. The suggestion to use the SPEM standard and the training in the modeling tool was one of the coaching activities of veteran students.

After this activity was carried out, during class #9 the students made a new reflection based on the acquired learning in the practical project. At the end of the teaching of the topics of this knowledge unit, the students answered the questionnaires about the knowledge acquired and developed new data dictionaries and concept maps. Figs. 7 and 8 present, respectively, an example of a conceptual map and data dictionary of the Software Process unit developed by the same student who made the map and dictionary showed in the Figs. 5 and 6.

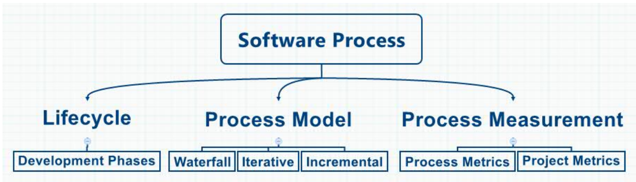

Fig. 7. Example of Conceptual Map (Post-Course)

Cascade: *Classic process model, which aims to develop in sequence, going through each stage individually.*

Lifecycle: *Development stages of a software system.*

Development Phases: *Steps to be followed during the use of a development model.*

Incremental: *A model in which the stages of a process are integrated and developed in parallel.*

Iterative: *Software process that aims at the repetition of certain phases of the project, aiming at the adequacy of the requirements.*

Process Measurement: *Metrics used to determine the duration, feasibility and suitability of a process.*

Process Metrics: *They give an idea of the effectiveness of an existing process.*

Process Models: *A set of process development techniques.*

Fig. 8. Example of Data Dictionary (Post-Course)

In order to evaluate the learning effectiveness of topics, we also considered the delivery of work products (requirements list, screen prototypes, source code, among others), the individual participation of students and the accomplishment of contextualized tests (with subjective questions regarding to the developed project).

IV. Analysis Results and Discussion

In order to organize the results obtained, the analysis of data on the learning of topics from the 6 knowledge units addressed is presented in Subsection IV-A. In addition, a discussion on the adoption of training strategies adopted in this experience report is presented in Section IV-B. We summarize the lessons learned in Section IV-C.

A. About the Learning Effectiveness of SE Topics

A questionnaire on the learning of SE topics at the beginning and, again, at the end of the course (the same questionnaire) was applied in order to evaluate the acquired learning. There was a considerable increase in the degrees "I know moderately" (41% in relation to prior knowledge) and "I know a lot" (20% in relation to previous knowledge). In addition, there were no answers "I know absolutely nothing" after the completion of the course.

This same questionnaire was applied with the class of veteran students, who completed the IPSE course in 2014 from a traditional teaching approach. Thus, from the comparison of the obtained results, a variation of 11% in the learning of the group was observed that followed industry training practices in relation to the group of veterans that followed a traditional approach, according to the graph of Fig. 9. In addition, the gain of the participating class of the experiment was 30% in relation to its initial knowledge.

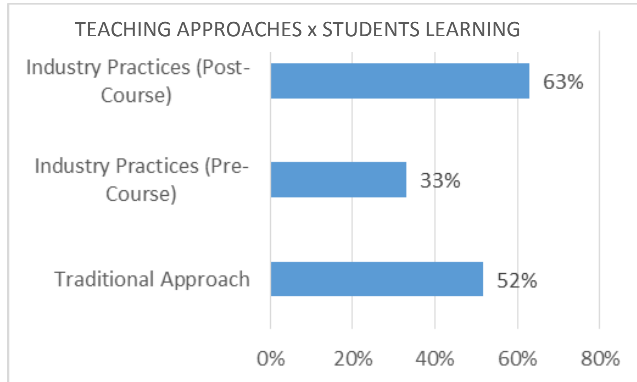

Fig. 9. Percentage of Student Learning in the IPSE Course

In order to qualitatively evaluate this students' learning, a subjective analysis was performed from the delivered work products (in a group). Thus, we observed an average of 6.9 pts in the grades obtained for these products. In addition, the lowest note was 6.0 pts and the highest was 8.0 pts. Regarding the contextualized tests (individual), the mean was 6.5 pts, the lowest being 5.3 pts and the highest 8.0 pts.

By the averages obtained, it is observed a greater difficulty to evaluate students based on the generation of products and involvement in the project. On the other hand, it was observed a greater understanding of the studied topics, due to the association of the evaluations with the developed software project.

B. About the Training Strategies

The most used training practice was mentoring, which consists of the consultant orienting and sharing with the professionals of the target company of the improvement, their experiences and knowledge in the SPI processes. This practice was adapted through the role of the professor, who acted as a mentor, maintaining and accompanying educational progress in the classroom. For assume this role, it is necessary for the professor to ask pertinent questions and present students with possible solutions. Thus, students choose which solutions will be adopted so that they can learn according to these choices.

The second most practiced was workshop, where the consultants hold a short seminar, presenting techniques and skills and demonstrating how they can be applied. This practice was adapted through the invitation to professionals and professors who work in certain knowledge areas of SE. They could contribute sharing their experience about difficulties and possible technical solutions related to this area.

We observed from the experience that the use of group dynamics has a great potential to involve the students and to explore the specific SE topics. For this practice, one difficulty was to identify dynamics suggestions for each knowledge unit.

In relation to the coaching practice, the experience was successful, being a positive point reported by the students and the professor. This practice was carried out by veteran students (monitors of the course), who carried out training in a certain technique or support tool.

Regarding the evaluation, we observed that if it is based on the delivery of work products it increases the engagement of the students in the practical project of the course. However, the individual participation of each student in the teams should be evaluated. Therefore, they can obtain an adequate grade for their contribution to the project. This contribution can be evaluated according to the percentage of individual participation of the students, which can be evaluated by all the team members through the following questions:

- Question 1: From 0 to 100%, what is the percentage of activities that your teammate performed (according to the responsibilities of his role)?
- Question 2: From 0 to 100%, how do you evaluate your colleague's participation (according to his contribution to the team)?

Finally, with regard to the contextualized tests we suggest that the professor elaborate subjective questions according to the knowledge units taught and contextualized with the activities developed in the practical project of the course. An example of a question might be: "Describe the process of collecting requirements from your project, highlighting the techniques used, generated documents, and requirements elicitors."

C. *Lessons Learned*

According to Gnatz et al. [12], the authors consider that it is important not only to theorize about the SE topics, but also to put them into practice. In this sense, they consider that it is possible to adapt certain industry practices to the context of a SE course. However, there is a shortage of qualified professors in academia who have actually acted in the industry [13].

In addition to the adoption of mentoring and workshops, which are practices more consolidated in the industry, we emphasize that the use of group dynamics, where an instructor suggests an activity analogous to a technical activity, and the practice of coaching, where a professional performs training in specific techniques, are still few explored by consultants and professors. We highlighted the need to adopt these practices, due to the inherent benefits of implementing these practices.

According to Prikladnicki et al. [3], the participants of group dynamics report that the interaction between the processes is better understood than in theoretical expositions. In addition, they allow increasing the level of interaction between the students and the facilitator, besides helping in the consolidation of the concepts from the experience of the theory. The practice of coaching, allows a professional specializing in certain processes to carry out a specific training, directly to the student. In this sense, coaching may be the most effective practice from a technical point of view.

Finally, regarding the evaluation practices, there was an increase in the quality of the work products delivered, besides the increase of the students' commitment with the practical project development of the course. In addition, we observe that subjective tests allow a more qualitative evaluation of learning of the knowledge units taught, from the contextualization of the topics application in the technical activities developed in the practical project of the course.

V. THREATS TO VALIDITY

The main limitation of this work was the number of participants in the survey. A low sampling rate was obtained, considering that the survey forms were released to 40 professors / consultants identified on the SOFTEX website (*www.softex.br/mpsbr*). However, in order to reduce this sampling bias, a survey was carried out in the northern, northeastern and southeastern regions of Brazil, representing around 80% of the population interviewed in the survey [8].

In total, we obtained answers from 5 SPI Implementing Institutions. We also highlight the average experience of the 10 consultants / professors interviewed, being 13 years and a half teaching and 11 years working with SPI consulting.

In addition, the sample size (14 students) of this experiment tends to weaken the external validity of this study, since the results obtained on these students may not be applicable to other educational institutions. Although, an individual experience may allow to a thorough analysis of the application of practices, the validity of its results depends on a broader empirical induction. In this sense, we intend to carry out a confirmatory case study, from the replication of this experience in other educational institutions in order to verify if the results will reinforce or weaken the results obtained in this paper.

VI. CONCLUDING REMARKS

A. *Contributions*

In the industry context, it is important that the people responsible for carrying out the processes to be deployed are not only important for the implementation of an SPI initiative, but also for its continuity [15]. More specifically, the relevance of this research is in addition to the efforts of the Brazilian industry to develop software products with the quality expected by the users and, consequently, to become competitive in the national and international markets.

In the academic context, the relevance given to the applicability of the proposal stands out. It is necessary to identify the contents that enable students to develop these professional competences [16]. The approach presented in this paper seeks to identify relevant topics for the professional of the area and to teach them in the SE course, in order to reduce the number of topics and focus on the most appropriated development of certain professional competencies and abilities.

In addition, the differential of this proposal is related to the use of training strategies and learning evaluation of the software industry, stimulating a more intense interaction between academia and industry. In this way, we expect that both the needs and the training of these two segments will be better aligned.

B. Future Works

The main contribution of this work is to provide the academic community of Software Engineering with a set of practices that can be adapted to the academic context in SE teaching. It is hoped that students will be able to obtain a more adequate training, from the adoption of training practices applied in the industry that allow the development of specific skills and abilities in Software Engineering. For professionals in the market, the contribution of this research will be the insertion of professionals better prepared to meet the demands of the market in the long-term.

As future work, from the results obtained in this experiment, it is intended to incorporate the main practices in a teaching framework. In this framework, the continuous representation strategy of the CMMI-DEV [10] will be adapted; so that professors and students can define which knowledge units will be the focus of course. Thus, the competencies and abilities of these selected units can be better developed.

Additionally, it is intended to carry out a case study of the application of these practices in other educational institutions, based on the feedback obtained in this initial experience. Thus, it will be possible to verify if similar results will be obtained.

ACKNOWLEDGEMENTS

The authors would like to thank the Coordination for the Improvement of Higher Education Personnel (CAPES) for the financial support to the development of this research.

REFERENCES

[1] ACM/IEEE, Computer science curricula 2013 – Curriculum guidelines for undergraduate degree programs in Computer Science, 2013.

[2] SBC, Curriculum guidelines of the SBC for for undergraduate degree programs in Computer Science and Computer Engineering, 2005.

[3] R. Prikladnicki, A. Albuquerque, C. Wangenheim, and R. Cabral, "Software Engineering Teaching: Challenges, Teaching Strategies and Lessons Learned", Proceedings of the II Software Engineering Education Forum, 2009. Fortaleza, Brazil.

[4] B. Bessa, M. Cunha, and F. Furtado, "ENGSOFT: Tool to Simulate Real Environments to Assist Problem-Based Learning (PBL) in Teaching Software Engineering", Proceedings of the XX Workshop on Computer Education, 2012. Curitiba, Brazil.

[5] C. Wangenheim, and D. Silva, "What Knowledge of Software Engineering is Important for a Software Professional?", Proceedings of the II Software Engineering Education Forum, 2009. Fortaleza, Brazil.

[6] S. Meira, "Information Systems and Software Engineering - Where are the Schools?", SBC Software Engineering Magazine - What is the impact of SE on the Computing market and on society as a whole?, 1st. ed. Chap. 1, 2015, pp. 11-15.

[7] I. Gimenes, "The Dilemmas of Software Engineering", SBC Software Engineering Magazine - What is the impact of SE on the Computing market and on society as a whole?, 1st. ed. Chap. 1, 2015, pp. 21-25.

[8] C. Portela, A. Vasconcelos, and S. Oliveira, "Analysis of the Relevance of Topics and the Effectiveness of Approaches for the Teaching of Software Engineering: Results of a Survey with Professors and Students", Proceedings of the VIII Software Engineering Education Forum, pp. 24-35, 2015. Belo Horizonte, Brazil.

[9] D. Nunes, C. Reis, and R. Reis, "Education in Software Engineering. The researcher's career in Software Engineering: principles, concepts and directions", 1st. ed. Chap. 5, 2010, pp. 132-181.

[10] SEI – Software Engineering Institute, "CMMI for Development – V 1.3", 2010. Avaliable in http://www.sei.cmu.edu/reports/10tr033.pdf

[11] SOFTEX – Association for the Promotion of Excellence in Brazilian Software "General MPS Software Guide", 2016. Avaliable in http://www.softex.br/wp-content/uploads/2016/04/MPS.BR_Guia_Geral_Software_2016-com-ISBN.pdf.

[12] M. Gnatz, L. Kof, F. Prilmeier, and T. Seifert, "A Practical Approach of Teaching Software Engineering", Proceedings of the 16th Conference on Software Engineering Education and Training, pp. 120-128, 2003.

[13] K. Garg, and V. Varma, "Software Engineering Education in India: Issues and Challenges", Proceedings of 21st Conference on Software Engineering Education and Training, pp. 110-117, 2008. Charleston, USA.

[14] S. Miranda, F. Ferreira, E. Costa, and F. Costa, "Experiment of a forum tool based on conceptual maps: a new form of interaction", Proceedings of XXVI Brazilian Symposium on Informatics in Education, pp. 917-926, 2015. Maceió, Brazil.

[15] A. Rocha, "GT/T4: Improvements in People Empowerment. Panel Improvements in the MPS.BR Program and in the MPS Model", Proceedings of XIII Brazilian Symposium of Software Quality (SBQS), pp. 9, 2014.

[16] D. Nunes (2015) "Software Enginner Competencies", SBC Software Engineering Magazine - What is the impact of SE on the Computing market and on society as a whole?, 1st. ed. Chap. 1, 2015, pp. 16-20.