Collaborative Strategy for Software Engineering Courses at a South American University

Miguel Alfonso Feijóo-García[1][a], Helio Henry Ramírez-Arévalo[1][b]
and Pedro Guillermo Feijóo-García[1,2][c]
1Program of Systems Engineering, Universidad El Bosque, Bogotá, Colombia
2Department of CISE, University of Florida, Gainesville, FL, U.S.A.

Keywords: Collaborative Learning, Peer-instruction, Software Engineering Education, Inter-curricular.

Abstract: Software Engineering (SE) is the discipline that integrates theory, methods, and tools to promote the development of new informatic solutions for multiple contexts. The discipline is generally introduced in Computer Science (CS) programs between the sophomore and junior years, adding the human being as an actor who participates in teamwork strategies to optimize time and effort. We report on an inter-curricular collaborative instructional strategy between two subsequent SE core courses—SE1 and SE2, at Universidad El Bosque, Colombia. We evaluated our strategy considering students' performance and perceptions, basing our analysis on their grades, Likert scale (1-5) responses, and the sentiment of their open-ended feedback— we calculated it with Natural Language Processing (NLP) techniques. Our findings suggest that an inter-curricular strategy like the one we present can foster students' performance, engagement, and motivation. Moreover, the strategy supports the promotion of SE skills, such as communication and teamwork.

1 INTRODUCTION

Software Engineering (SE) is the discipline that gathers the theory, methods, and tools used in processes involving the development of new informatic solutions (Somerville, 2020). This discipline invites to go beyond technical components to promote systemic thinking in business contexts. Some of the perspectives promoted by SE are 1) Methodological: how to optimize human and technological resources in a software development process, 2) Design and Modeling: how to optimize the structure and dynamics of the systems to be designed, and 3) Technological: how to gather existing technologies in the design of solutions to contextual problems (i.e., companies, individuals, and societies). Hence, promoting structured and systemic thinking skills required by this discipline, implies various educational challenges from a holistic perspective.

At Universidad El Bosque, Colombia, we lead our students' professional development following the structure proposed by the Biopsychosocial & Cultural Model (BPsy&C). The BPsy&C proposes four dimensions based on a perspective centered in 1) the environment, 2) the artifact, 3) the habits, and 4) the beliefs. This model fosters the development of a global analysis in the context of a certain project, multi-disciplinarily helping in the understanding and enhancement of complex needs (López-Cruz & Ortíz-Buitrago, 2017).

SE requires of teaching-learning processes to be incremental and evolving, based on curricular approaches that gather previous skills and knowledge from prior courses (e.g., CS1, CS2, Data Structures). Software development does not just depend on the technology used (e.g., third-party tools, context-based components), but also on the methodologies that lead to good practices in the management of human resources, time, among others. Based on the constructivist theory (Saldarriaga-Zambrano *et al.,* 2016), we need to foster teaching-learning processes to be based on the construction of knowledge (i.e.,

[a] https://orcid.org/0000-0001-5648-9966
[b] https://orcid.org/0000-0001-6420-5687
[c] https://orcid.org/0000-0002-3024-1257

mental models) from enriching experiences, further from the basic transmission of concepts or topics.

The integration and relationship of CS courses through transversal activities promotes a holistic professional development that bridges concepts and skills coming from different courses. Nevertheless, we have observed at our institution that SE courses, regardless of belonging to the same curricular line, do not fully satisfy the integration of strategies: there is a particular interest in micro-curricular topics. Thus, we ask the next questions: How do we guarantee that SE students link competencies from different courses? How do we promote a clear learning roadmap to our students from each one of the SE courses? How do we make this roadmap to be learner-centered?

This paper presents a strategy applied between two SE core courses on subsequent semesters (SE1 and SE2), using a transversal project that demanded competencies from both courses. Our strategy fostered the use of good software development practices, asking students to use agile methodologies that promoted inter-curricular teamwork and helped them build skills on requirements identification, responsibilities delegation, and decision-making: skills required in Industry. We present our findings and results on students' perceptions based on our active learning approach, and the impact of our strategy for their learning processes. Additionally, we describe the instructors' experiences, addressing the pros and cons from this process.

2 RELATED LITERATURE

Software Engineering Education (SEE) has been explored in Computing Education Research (CER) for decades, concerning topics such as 1) software development processes, 2) software modeling, and 3) collaborative learning.

In recent years, the CS community has advocated on agile methodologies in CS curricula, referring to its benefits compared to traditional waterfall approaches, as also as their contribution to the professional development of Computer Scientists (Soundararajan *et al.,* 2012; Soundararajan & Arthur, 2012; Campanelli & Parreiras, 2015; Tripp & Armstrong, 2018). This has motivated the CS Ed community to brainstorm and evaluate novel teaching-learning strategies to introduce these methodologies. Literature exists on game-based activities for requirements definition (Beatty & Alexander, 2008; Knauss *et al.*, 2008; Hof *et al.*, 2017), assignments using LEGO as a tool to teach

methodologies (Kurkovsky, 2015; Kurkovsky *et al.*, 2019), and games designed to assist learners in SE: board games (Brito & Vieira, 2017; Moura & Santos, 2018) and digital ones (Marinho *et al.*, 2020; Rodriguez *et al.,* 2015). The CS Ed community has also contributed to strategies to help students learn about software modeling and software design (Pérez & Rubio, 2020; Gayler *et al.*, 2007; Coffey, 2017). Technologies such as DesignDB (Goelman & Dietrich, 2018) and Archinotes (Urrego *et al.,* 2014) pose as examples of tools designed to leverage software abstraction and modeling for SE.

Finally, collaboration plays an essential role in SE. CER literature reports that using strategies like pair-programming positively impact intra-curricular CS setups. Students who participated in Collaborative Learning (CL) activities such as pair-programming improved their learning performance (Gray *et al.*, 2019) as they also increased their confidence (Celepkolu & Boyer, 2018). These outcomes relate to research on inter-curricular CS setups between first-year courses: CS1 and CS2 (Feijóo-García & Ortíz-Buitrago, 2018; Cottam *et al.*, 2011). Like pair-programming, peer-tutoring reported helping improve students' performance, retention, and motivation: primarily, when students attributed a mentor role. Additional literature presents Global Software Engineering (GSE). GSE has taken place in undergraduate and graduate courses all around the world. As it reports about the benefits of CL, it also presents challenges due to cultural and language barriers regardless of the configuration between institutions (Fu *et al.*, 2018; Clear *et al.,* 2015).

Our work contributes to CER literature on SE with a strategy based on an inter-curricular design between two subsequent SE courses in the same institution. The strategy uses the benefits of peer instruction, addressing SE concepts and skills concerning methodology, project management, and software design.

3 CONTEXT AND STRATEGY

This section describes the courses: SE1 and SE2, in which we used our active-learning strategy. Additionally, we present its evaluation and how we carried out the data analysis to get the findings and results we explain later in this paper.

Our strategy is carried out in two mid-undergraduate core (i.e., mandatory) courses in the Program of Systems Engineering— i.e., Computer Science (CS), at a South American higher-education institution—Universidad El Bosque, Colombia. SE1

is offered to sophomore CS students. This course introduces defined structures for the design and development of team software projects, using reference frameworks on traditional and agile methodologies (i.e., TSP, RUP, Scrum, XP). On the other hand, SE2 is offered to junior CS students. This course addresses topics related to software patterns and architecture (i.e., Observer, Factory, Facade, MVC, SOA, MSA), software quality, metrics, software estimation, usability, and distributed software. At this mid-level point of our students' professional development, they already have gained concepts and skills on CS1, CS2, Data Structures (CS3), Algorithms Design (AD), and Databases (DB). Hence, SE1 and SE2 seek to improve skills and abilities following the complete software development life cycle, systems modeling, and the administration of a software development project in business environments.

The College of Engineering of our institution divides each course into three academic modules. Each course has a duration of 16 weeks (i.e., academic semester in Colombian standard). Throughout the semester we proposed two projects, aiming to apply the topics carried out in each course: SE1 or SE2. The first project started in the second half of the first module to the end of the first half of the second module, with a duration of four weeks. We proposed the second project to last five weeks, during the third academic module of the semester. We had a total of 36 students (N=36): 50% (n=18) from SE1, and 50% (n=18) from SE2. We had 13.88% (n=5) female students, and 86.11% (n=31) male students. We had students between 18 and 41 years of age: 69.4% (n=25) between 18 and 21 years of age, 25% (n=9) between 22 and 25 years of age, and 5.6% (n=2) over 25 years of age.

3.1 First Project Approach [Intra-curricular]

This subsection describes the first context-based project, which made use of an intra-curricular design to promote collaborative learning. The project's context was the same for SE1 and SE2. However, we asked some specific deliverables and tasks for each course, depending on the topics seen to date.

In this first project of the semester, we formed groups of six people within each course (SE1 or SE2), to work on a web-based software solution according to the topics carried out at the time. We formed six working groups: three groups from SE1 and three from SE2. Groups were asked to develop web-based software solutions using a traditional waterfall

methodology (i.e., RUP, TSP). This project was centered on software solutions for a national-wide movie theater company. The software development process promoted intra-curricular interactions between students at the same academic level, and fostered teamwork skills and responsibilities' delegation. Moreover, each group had to use all the concepts seen to date in each course.

After the groups' completion of the software development process, we proceeded with an inter-curricular peer-reviewing approach. We selected three members from each group in each course (SE1: n=9, SE2: n=9). Groups in SE2 were reviewed by SE1 students, as groups in SE1 received feedback from SE2 students. Each reviewer was asked to solely evaluate one group. This approach helped us to explore how an inter-curricular design could work between SE1 and SE2.

3.2 Cross-sectional Project Approach [Inter-curricular]

This subsection describes the second context-based generic cross-sectional project. The project's context was the same for SE1 and SE2. However, differently from the first project, this second project's design was inter-curricular between SE1 and SE2 for the software development process.

In this second project of the semester, we formed groups of six people between both courses (SE1 and SE2), to work on a web-based software solution according to the topics carried out at the time. We formed six inter-curricular working groups. Groups were asked to develop web-based software solutions using an agile methodology (i.e., Scrumban, Scrum, XP, LSD). This project was centered on software solutions for a national-wide parking management company. The inter-curricular design for the software development process promoted teamwork skills and responsibilities' delegation, considering different levels of expertise between students. Additionally, our approach fostered a collaborative learning environment that contributed to both kinds of students: SE1 students were introduced to new concepts common in SE2, while SE2 students reinforced previous concepts and skills from SE1.

We had two clients and each of them were assigned to three groups—first and second authors of this paper. As clients, we monitored each group's development, progress, and evolution, both documentary and technologically. After the groups completed the second project (five weeks), we evaluated each solution through a formal presentation (i.e., postmortem) asking for the required

deliverables: e.g., Software Architecture Document (SAD), video-demo, web-based software—deployed in a cloud platform. During these presentations, we provided feedback on groups' decisions, outcomes, and teamwork process. We also provided a survey where our students were asked to self-evaluate and co-evaluate according to their contributions. Students also responded to a survey to assess the activity's effectiveness based on their perception, reporting on the benefits, difficulties, and impact of the project and the proposed strategy.

Differently from the first project, we had to work for this second project online. This, due to limitations because of COVID-19. The pandemic brought limitations in terms of communication, software development, and teamwork, in addition to the stress of the pandemic. However, the use of new alternative resources for synchronous and asynchronous teamwork allowed our students to gain new skills on time management and usage of resources. Moreover, the inter-curricular design helped them to keep engaged and motivated. We first thought that students were not going to respond successfully to our approach due to the pandemic. However, their participation was active, and the outcomes were satisfying.

4 DATA ACQUISITION

We evaluated the effectiveness of the strategy mentioned above considering the following aspects: 1) students' submissions, reviewed by the instructors and their peers [quantitative— ratio data from 0.0 to 5.0], and 2) the students' perceptions on their experience with the strategy [quantitative—1-5 Likert scales, and qualitative—open-ended questions]. In this section, we present the data acquisition for each aspect.

4.1 Data Acquisition: Evaluation Phase

For each project, we asked our students to create a presentation, in addition to a documentation on their software development process. Both projects were graded on a scale between 0.0 and 5.0 and had three components—the leading instructors determined the percentage weights based on their experience with both courses. The evaluation criteria considered:

Presentation—30% of the Project's Grade: We evaluated the presentation's content, the number of functionalities developed for the web-based software, the rationale behind the software development process, their communication skills, and their

responses to their observations and questions posed by their instructors.

Documentation—50% of the Project's Grade: We asked students to document their process and results on a software architecture document (SAD), in addition to a test-planning document, and two manuals: a technical one, and a usability one—this one included a video-demo. Documentation was evaluated by their instructors.

Peer-reviewing—20% of the Project's Grade: Peers were asked to review their teammates considering time management, effort, and engagement with the software development process.

4.2 Data Acquisition: Perceptions

We gathered our students' perceptions on the second project (i.e., inter-curricular) with a questionnaire that had five Likert-scale (1-5) questions and three open-ended questions (see Table 1).

Table 1: Questionnaire for our student's perceptions.

Question	Option
Q1: If you had to evaluate this strategy, with its methodology, advantages, disadvantages, opportunities and difficulties, how **useful** would you find it?	$(1-5)$ 1: Not useful at all. 5: Very useful.
Q2: Indicate how **comfortable** you felt with this inter-curricular strategy that involved two courses from different academic semesters.	$(1-5)$ 1: Very uncomfortable. 5: Comfortable.
Q3: Indicate how much **effort** did you have to invest in for this inter-curricular strategy.	$(1-5)$ 1: No effort at all. 5: Much effort.
Q4: Indicate how much did the inter-curricular strategy **contribute** to your professional development.	$(1-5)$ 1: No contribution at all 5: It contributed very much.
Q5: Indicate the **impact** of the inter-curricular strategy for your professional development.	$(1-5)$ 1: No impact at all. 5: It impacted very much.
Q6: Indicate the **positive aspects** of the inter-curricular strategy you were asked to follow.	Open-ended question
Q7: Indicate the **difficulties** of the inter-curricular strategy you were asked to follow.	Open-ended question
Q8: Briefly justify your previous answers. All comments, reflections, and perceptions must be recorded in this section.	Open-ended question

5 FINDINGS AND RESULTS

We present our findings and results based on the data acquired with the instruments described in section 4, focusing our analysis on each of the data acquisition categories previously described.

5.1 Data Analysis: Evaluation

Looking at Table 2, we can observe that the scores of the quantitative evaluations, given by the instructors (80% of the final score) and peers (20% of the final score), were generally positive. Each group got an average score higher than 4.0 in a 0.0 to 5.0 scale. Each group's score was given based on their submission and the process they reported. For the inter-curricular project—first project, scores were normally distributed (Shapiro-Wilk, w=0.95, p >0.05): 47% of students were scored higher than 4.0 (n=17), 50% of students were scored lower than 4.0 and higher than 3.0 (n=18), and 3% of students were scored lower than 3.0 (n=1). For the inter-curricular project—second project, scores were not normally distributed (Shapiro-Wilk, w=0.89, p < 0.01). This, due to students' performance on the inter-curricular project: 56% of students were scored higher than 4.0 (n=20), and 44% of students scored lower than 4.0 and higher than 3.0 (n=16).

We had positive results for both projects and methods. Moreover, based on the scores' distributions mentioned above and the descriptive statistics presented in Table 2, we can suggest that the inter-curricular design helped students to get better scores. However, further research should be conducted to validate that claim.

Table 2: Descriptive Statistics on Students' Scores.

Project	Course	Mean	SD	Median
Project #1 – intra-curricular	SE1	3.87	0.62	3.81
	SE2	4.10	0.62	4.33
Project #2 – inter-curricular	SE1	4.02	0.42	4.14
	SE2	4.02	0.42	4,14
	SE1 & SE2	4.02	0.42	4.14

5.2 Data Analysis: Perceptions

We did an analysis on the Likert scales (1-5) (Joshi *et al.,* 2015) used to gather students' perceptions (see Table 1). This analysis is represented as a divergent stacked-bar graph (Tufte & Graves-Morris, 1983), and it helped us to identify how effective did students perceive the proposed inter-curricular project and its methodology (Fig. 1). Additionally, with natural language processing techniques (NLP) of students' comments—Naïve Bayes classification technique (Jurafsky & Martin, 2014), we were able to analyze text sentiment on their input, and to create word clouds based on their responses to questions 6, 7, and 8 (see Table 1)—i.e., most highly mentioned words. This analysis guided us to reflect on the positive aspects and difficulties perceived by our students, helping us in the identification of elements to improve for our inter-curricular strategy.

As presented in Figure 1, students from both courses (SE1 and SE2) generally considered the inter-curricular strategy "Satisfactory" (n=9) or "Very Satisfactory" (n=22). Our findings suggest that SE1 students benefitted the most from our strategy due to the interaction they had with SE2 students, as also due to the introduction of upcoming SE2 topics. However, SE2 students' responses differed on Q2 and Q4 (see Figure 1). We believe that it is due to the difficulty SE2 students found when they assigned responsibilities according to SE1 peers' skills at the beginning of the project. However, further research is required to understand those perceptions.

We conducted a sentiment analysis using semantic NLP on our students' comments—Naïve Bayes classifier (Jurafsky & Martin, 2014) with TextBlob (Loria, 2018). We distributed our utterances in three categories: Positive, Neutral, and Negative. Table 3 presents the performance of the Naïve Bayes model's accuracy. For this, we used a N=44 training set.

Table 3: Naïve Bayes Model on Sentiment Classification.

Category	Precision	Recall	F1-Score	Support
Positive	0.67	0.92	0.77	13
Neutral	0.73	0.73	0.73	11
Negative	1.00	0.75	0.86	20
Accuracy Calculations				
Accuracy	-	-	0.80	44
Average (Macro)	0.80	0.80	0.79	44
Average (Weighted)	0.83	0.80	0.80	44

The accuracy of the model used to classify students' comments was 80% (Table 3). We can affirm that the results obtained from the comments of our students are reliable, based on Díaz et al. contribution: recent studies published in the academic community Teaching Academic Survival Skills (TASS), present accuracy values between 63.1% and 89.3% on sentiment-based classifiers (Díaz-Galiano *et al.,* 2019). Table 4 presents the results obtained from our sentiment analysis per category.
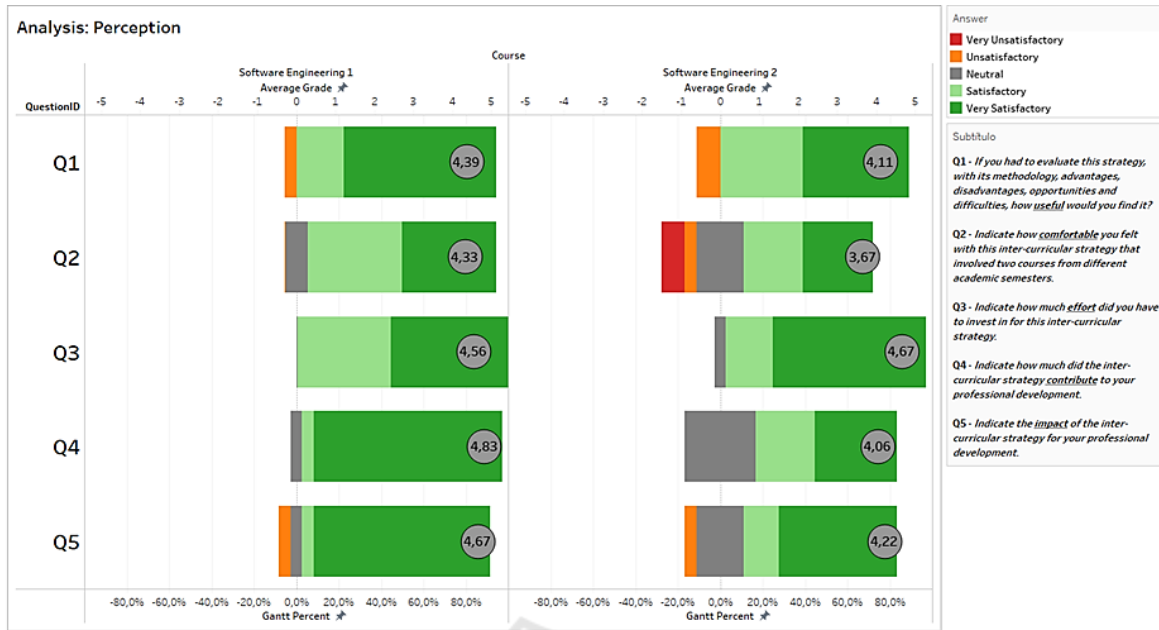
Figure 1: Likert-Scale (1-5) visualization on open-ended questions about student's perceptions.

Table 4: Sentiment Analysis per category.

Cat/Aspect	Pos.	Diff.	Perceptions	Pct (%)
Positive	23	20	30	67.6%
Neutral	6	5	2	12.0%
Negative	7	11	4	20.4%

For the inter-curricular project, students generally responded with positive perceptions. This, since 7 out of 10 students (67.6%) made positive comments on Q6, Q7, and Q8 (see Table 1): 1) Positive Aspects, 2) Difficulties, and 3) Perceptions. Moreover, responses on Q6, Q7, and Q8 were in average positive—61% of students. On the other hand, the percentage of positive responses regarding the inter-curricular project's general perception was 77%, and most of the comments (55.56%) were positive even in terms of those difficulties students identified.

The words most frequently used by our students per question (Q6, Q7, and Q8) were: (1) Difficulties: time, communication, and difficulty, (2) Perceptions: Group work, good experience, knowledge, and (3) Positive Aspects: Learning, knowledge, group work. Regardless of the existing limitations, the inter-curricular strategy had a positive general perception for our students. We believe that students highly appreciated the team-based design, finding our approach as a pleasant learning experience. This claim is based on the sentiment analysis we have described.

6 DISCUSSION

We consider that our strategy effectively assisted in the development of soft skills (e.g., communication, teamwork, assignment of responsibilities, resource management), and allowed students to understand concepts and gain skills from topics from both courses. This activity required us to invest additional effort as instructors to supervise each team upon their expected development process. Regardless of the cost of planning meetings with the different teams to evaluate their progress and outcomes, we find very satisfying how our students engaged, and the motivation they exhibited with the proposed strategy.

We found two aspects we consider were difficult to address: (1) At the beginning, SE2 students misinterpreted the assignment's objective. They believed that their role was to instruct SE2 topics to SE1 peers. As instructors, we had to clarify that the strategy was asking them all to work as peers, as their goal was to guarantee the best responsibilities' assignment and distribution according to their skills. We believe that the misinterpretation was due to the lack of inter-curricular strategies. However, we consider that it was not something that impacted the later steps in our strategy. (2) Although both projects had minimum requirements, there were some additional features asked regarding each group and their processes. We found easy to evaluate the minimum requirements between groups, but we had

to invest extra time to scale and grade those additional features requested per group. We will standardize features for upcoming iterations of our strategy.

We also find that the students' comments on the activity were positive and constructive, and that they guide us to improve our strategy for future iterations. We found that when the activity was first proposed, students were reluctant to work with peers who did not belong to their same course (SE1 or SE2). However, after starting our strategy, we observed our students committing to the software process and engaging with their peers. This shows us that our strategy was beneficial to foster and develop Software Engineering skills.

As educators, we cannot ignore the opportunity to highlight this experience and the satisfaction that our strategy gave us. We consider that the teamwork, attitude, assimilation, and motivation we observed in our students were positive. Additionally, our inter-curricular strategy fulfilled its goal, by guiding our students to get the most out of it based on the concepts, skills, and competences expected in our Software Engineering courses.

ACKNOWLEDGEMENTS

REFERENCES

Beatty, J., & Alexander, M. (2008). Games-based requirements engineering training: an initial experience report. In 2008 *16th IEEE International Requirements Engineering Conference* (pp. 211-216). IEEE.

Brito, A., & Vieira, J. (2017). '2TScrum' A Board Game to Teach Scrum. In Proceedings of the *31st Brazilian Symposium on Software Engineering* (pp. 279-288).

Campanelli, A. S., & Parreiras, F. S. (2015). Agile methods tailoring–A systematic literature review. Journal of Systems and Software, 110, 85-100.

Celepkolu, M., & Boyer, K. E. (2018). Thematic analysis of students' reflections on pair programming in cs1. In Proceedings of the *49th ACM Technical Symposium on Computer Science Education* (pp. 771-776).

Clear, T., Beecham, S., Barr, J., Daniels, M., McDermott, R., Oudshoorn, M., ... & Noll, J. (2015). Challenges and recommendations for the design and conduct of global software engineering courses: A systematic review. In Proceedings of the *2015 ITiCSE on Working Group Reports* (pp. 1-39).

Coffey, J. W. (2017). A study of the use of a reflective activity to improve students' software design capabilities. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 129-134).

Cottam, J. A., Menzel, S., & Greenblatt, J. (2011). Tutoring for retention. In Proceedings of the 42nd *ACM technical symposium on Computer science education* (pp. 213-218).

Díaz-Galiano, M. C., García-Cumbreras, M. Á., García-Vega, M., Gutiérrez, Y., Cámara, E. M., Piad-Morffis, A., & Villena-Román, J. (2019). TASS 2018: The strength of deep learning in language understanding tasks. *Procesamiento del Lenguaje Natural*, 62, 77-84.

F. Tripp, J., & Armstrong, D. J. (2018). Agile methodologies: organizational adoption motives, tailoring, and performance. *Journal of Computer Information Systems*, 58(2), 170-179.

Feijóo-García, P., & Ortíz-Buitrago, C. (2018). The Godparent Plan: A Pedagogical Strategy for CS1 Accompaniment and CS2 Pedagogical Enhancement.

Fu, Y., Reina, L. P., & Brockmann, P. (2018). Teaching Global Software Engineering: Experience Report Comparing Distributed, Virtual Collaborative Courses at the Bachelor's and Master's Degree Levels. In Proceedings of the 3rd *European Conference of Software Engineering Education* (pp. 34-38).

Gayler, D., Klappholz, D., Harvey, V. J., & Pérez-Quiñones, M. A. (2007). UML tools: what is their role in undergraduate computer science courses?. In Proceedings of the *38th SIGCSE technical symposium on Computer science education* (pp. 129-130).

Goelman, D., & Dietrich, S. W. (2018). A Visual introduction to conceptual database design for all. In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (pp. 320-325).

Gray, J., Haynie, K., Trees, F., Astrachan, O., Uche, C., Cooney, S., & Kick, R. (2019). Infusing Cooperative Learning into AP Computer Science Principles Courses to Promote Engagement and Diversity. In Proceedings of the *50th ACM Technical Symposium on Computer Science Education* (pp. 1190-1196).

Hof, S., Kropp, M., & Landolt, M. (2017). Use of Gamification to Teach Agile Values and Collaboration: A multi-week Scrum simulation project in an undergraduate software engineering course. In Proceedings of the 2017 *ACM Conf. on Innovation and Technology in Computer Science Education* (pp. 323-328).

Joshi, A., Kale, S., Chandel, S., & Pal, D. K. (2015). Likert scale: Explored and explained. Current Journal of Applied Science and Technology, 396-403.

Jurafsky, D., & Martin J.H. (2014). Speech & language processing. Upper Saddle River, NJ: Prentice Hall, Pearson Education International.

Knauss, E., Schneider, K., & Stapel, K. (2008). A game for taking requirements engineering more seriously. In 2008 *Third International Workshop on Multimedia and*

Enjoyable Requirements Engineering-Beyond Mere Descriptions and with More Fun and Games (pp. 22-26). IEEE.

Kurkovsky, S. (2015). Teaching software engineering with LEGO Serious Play. In Proceedings of the 2015 *ACM Conference on Innovation and Technology in Computer Science Education* (pp. 213-218).

Kurkovsky, S., Ludi, S., & Clark, L. (2019, February). Active Learning with LEGO for Software Requirements. In Proceedings of the *50th ACM Technical Symposium on Computer Science Education* (pp. 218-224).

López-Cruz, O., & Ortíz-Buitrago, C. H. (2017). Formación de ingenieros informáticos como gestores de innovación: transformadores de vidas, constructores de futuros posibles. ANFEI Digital, (7).

Loria, S. (2018). textblob Documentation. Release 0.15, 2.

Marinho, L. L., dos Santos, S. R. C., Andrade, L., Cons, B. C., Schots, M., & Werneck, V. M. (2020). Scrumie: Scrum Teaching Agent Oriented Game. Revista de Informática Teórica e Aplicada, 27(2), 140-152.

Moura, V., & Santos, G. (2018). ProcSoft: A Board Game to Teach Software Processes Based on ISO/IEC 29110 Standard. In Proceedings of the 17th Brazilian Symposium on Software Quality (pp. 363-372).

Pérez, B., & Rubio, Á. L. (2020). A project-based learning approach for enhancing learning skills and motivation in software engineering. In Proceedings of the 51st *ACM Technical Symposium on Computer Science Education* (pp. 309-315).

Rodriguez, G., Soria, Á., & Campo, M. (2015). Virtual Scrum: A teaching aid to introduce undergraduate software engineering students to scrum. Computer Applications in Engineering Education, 23(1), 147-156.

Saldarriaga-Zambrano, P. J., Bravo-Cedeño, G. D. R., & Loor-Rivadeneira, M. R. (2016). La teoría constructivista de Jean Piaget y su significación para la pedagogía contemporánea. Dominio de las Ciencias, 2(3 Especial), 127-137.

Sommerville, I. (2020). Software Engineering. 10th. In Book Software Engineering. 10th, Series Software Engineering. Addison-Wesley.

Soundararajan, S., & Arthur, J. D. (2011). A structured framework for assessing the" goodness" of agile methods. In 2011 18th IEEE *International Conf. and Workshops on Engineering of Computer-Based Systems* (pp. 14-23). IEEE.

Soundararajan, S., Chigani, A., & Arthur, J. D. (2012, February). Understanding the tenets of agile software engineering: Lecturing, exploration and critical thinking. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 313-318).

Tufte, E. R., & Graves-Morris, P. R. (1983). The visual display of quantitative information (Vol. 2, No. 9). Cheshire, CT: Graphics press.

Urrego, J., Muñoz, R., Mercado, M., & Correal, D. (2014, May). Archinotes: A global agile architecture design approach. In *International Conference on Agile Software Development* (pp. 302-311). Springer, Cham.