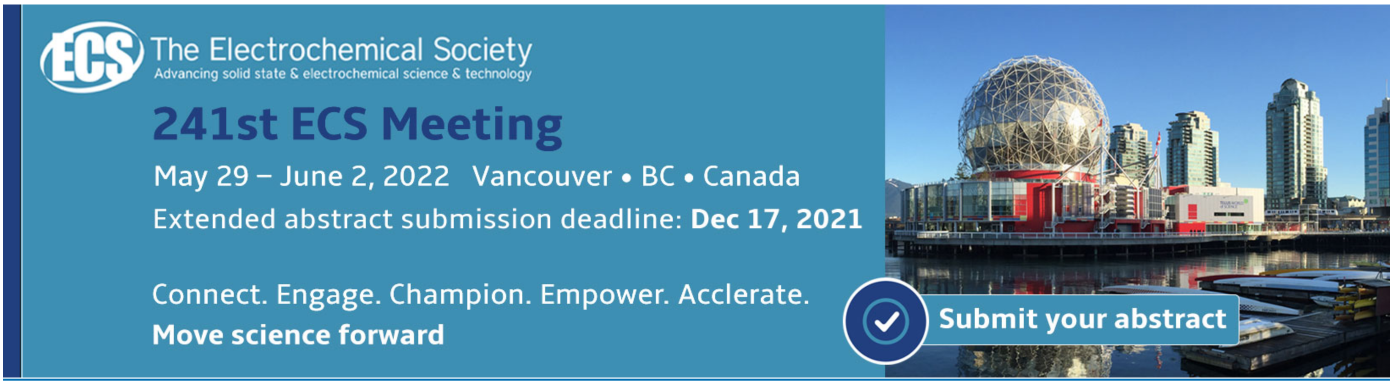
Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Measuring the quality of the blended learning approach to teaching computational sciences

To cite this article: B Perez *et al* 2020 *J. Phys.: Conf. Ser.* **1587** 012021

View the article online for updates and enhancements.



Measuring the quality of the blended learning approach to teaching computational sciences

B Perez[1,2], **C Castellanos**[2], **and D Correal**[2]

[1] Grupo de Investigación en Inteligencia Artificial, Universidad Francisco de Paula Santander, San José de Cúcuta, Colombia
[2] Grupo de Investigación en Tecnologías de Información y Construcción de Software, Universidad de los Andes, Bogotá, Colombia

E-mail: borisperezg@ufps.edu.co, cc.castellanos87@uniandes.edu.co

Abstract. Teaching models are a very interesting area of study that can be applied to any science. Finding the best way to transfer knowledge to students, so that they internalize it and can successfully transfer it to the field, is key to the country building. This article measures the quality of a teaching model that can be applied in physics or engineering, which requires a correct theoretical foundation before addressing practical elements. Blended learning is an approach to education that combines online educational materials and opportunities for interaction online with traditional place-based classroom methods. This model was applied in agile software architectures course and measured through an anonymous survey of students in software architecture, who were enrolled for the periods 2015-2018. The results showed that 39.7% of the respondents considered that they learned better compared to face-to-face courses.

1. Introduction

Software architecture is a key component of education. However, teaching it is a difficult task that requires overcoming challenges such as teamwork, technical and social skills, and realistic contexts [1–5]. This is why his teaching has received much attention in both academia and industry.

Several strategies have been suggested to improve the learning of software architecture, such as: inclusion of agile methodologies [6–8], inverted classroom [8–10], capstone projects [11–13], communication skills [3], comparison of data mining techniques [14], and architecturally savvy personas [8]. However, despite the attention surrounding technical debt (TD) by both the industry and academia, there is still a lack of empirical evidence about the effectiveness of these practices.

In the agile software architectures master course, the blended learning methodology was used. The course consists of six face-to-face classes, and nine virtual classes, focused on the design of a software architecture for a real industry problem, through the application of agile methodologies. But, it is also required to evaluate if this strategy was suitable for this course, and in general, for software architecture courses.

Given the aforementioned requirement, the goal of this study is to find out if the skills taught in class manage to stay with the students long after they have finished the course through a blended learning methodology. With this learning methodology, we seek to reinforce a set of skills that we believe a student of this course should develop: self-learning, self-criticism, teamwork, problem solving and professional development. The motivation behind this research is twofold: (1) to evaluate the student's perception of whether the course, in blended learning mode, supports the development of the skills we believe a software architect should have; and (2) to evaluate whether this model allows students to better learn the concepts taught in the agile software architecture course. For this, a survey was applied to all students who have taken the course. A total of 63 students responded to the survey. The results showed that 39.7% of the respondents considered that they learned better compared to face-to-face courses. Additionally, the most valued aspect of respondents was the use of industry problems.

Results showed that students agreed on the importance and satisfaction of using this blended teaching technique to learn about software architecture. However, its effectiveness could not be demonstrated. This could be caused by the few face-to-face classes of the course. These results could help software architecture instructors in selecting the best possible strategy to improve the teaching process. This way, on a bigger scale, this could be traduced to better future professionals to deal with current challenges in the software industry. Companies increasingly need qualified professionals, including software architects.

Following we describe the course design in section 2. We present the followed methodology for the survey in section 3. Our results are presented in section 4. Discussion and conclusions are presented in section 5.

2. Course design

In this section, we describe the context of the agile software architectures course using the following situational factors: general context, specific context, nature of the subject and characteristics of the learners.

- General context of the learning situation. The training of software architects is usually the result of several years of work experience and participation in information technology (IT) projects. To meet industry requirements, our university initiated a master's degree in IT architecture, and with it, the agile software architectures course. By the end of 2017, more than 180 students had graduated as IT architects.
- Specific context of the learning situation. This course in software architecture was designed with two main objectives: (i), following an agile approach and, (ii), implementing a blended learning methodology. The course is focused on designing the architecture of a software system as proposed by an actual company and with real stakeholders. Students implement five sprints during the semester, performing an incremental design of their architecture.
- Nature of the subject. During each sprint, students design for a particular quality attribute. This implies that they must read and investigate the different aspects necessary to solve a particular challenge. Once they have a design, critical decisions are tested, implementing and deploying prototypes to measure whether or not quality requirements are met.
- Characteristics of the learners. The objective public for this course is postgraduate IT students from both masters programs (IT architecture and software engineering), usually in their middle twenties with 4 years of experience as developers or software architects.

3. Methodology

According to Easterbrook, *et al.* [15], selecting a research method will depend of the research questions. Having selected them and considering the objectives of our study, we decided to use a survey research instrument.

3.1. Research questions

The paper aims to uncover if our blended teaching technique could be useful to improve the learning process of software architecture. This work is guided by the following research questions (RQ):

- RQ1: What is the students' perception of the agile software architecture master's course regarding the blended learning methodology?
- RQ2: How does the blended learning methodology allow students to better learn the concepts taught in the course?

3.2. Survey design

This survey collects individual opinions and preferences, and demographic information. It is classified as exploratory and descriptive research because it focuses on the discovery of perspectives, it was previously planned and structured, and the information collected can be inferred statistically about a population. In this survey, the target population was limited to professionals who took the agile software architecture course at our university. This survey employed a non-probability sampling method because the population was selected based on the subjective judgment of the researcher rather than random selection. Respondents were easily accessible to the researchers in this survey. Snowball sampling was not considered.

3.3. Creating the questionnaire

Survey questions were based on our two research questions (see section 3.1). This survey instrument was made up of 23 questions using Google forms: five closed-ended demographic questions and eighteen specific questions; where sixteen are closed-ended questions related to perception on blended learning methodology and two open-ended questions on general perception.

The protocol of this survey was first refined iteratively by two authors. Then, the questionnaire was reviewed externally by one professor in charge of the course, as well as two practitioners on software architecture. We refine the survey instrument by having a pilot study with five participants.

We conducted the survey in the period from August 28th, 2018 to September 12th, 2018. The survey was sent by email including a welcome message explaining the purpose of the survey, our affiliation and contact information. To keep the survey as simple as possible, we presented all the questions on a single page.

3.4. Data analysis

For closed-ended questions (quantitative answers) we used descriptive statistics. Answers for open-ended questions were analyzed using a code schema and extracting the main points to a Excel sheet.

4. Results

In this section we present the student perception related to our blended teaching technique based on the survey answers. The main survey findings are presented in Figures 1 to Figure 3.

4.1. Demographic data

We sent the survey to 227 students of the current and previous editions of the agile software architectures course. In total 63 responses were received. This represents 27.7% of response rate.

Related to their demographic information, most respondents (20 out of 63, 32%) were between 31 and 36 years old, followed by respondents between 26 an 30 years old (19 out of 63, 30%).

Related to their gender, vast majority of respondents were male (57 out of 63, 90.5%). With respect to their educational background, almost half of the respondents (31 out of 63, 49.2%) hold a bachelor of science degree. About their role in their companies, most respondents selected "development" (28 out of 63, 44.4%) as their main task, and "software architecture" follows behind with 12 (19%) respondents. Finally, related to the domain of the companies where respondents work, the vast majority of respondents reported working in "IT" or "technology" (24 out of 63, 38.1%). "Government" (7 out of 63, 11.1%) and "finance" (6 out of 63, 9.5%) were in second and third place respectively.

4.2. Students' perception through closed-ended questions

In general terms, students agreed on the importance and satisfaction of using this methodology to learn about software architecture (Figure 1). This agreement is evidenced by 46 positive responses (strongly agree: 23; agree: 23). The effectiveness of this blended learning methodology was the only question with a lower median than the other answers. Students prefer to be in constant interaction with the professor in order to clarify issues or have a better understanding of the concepts.

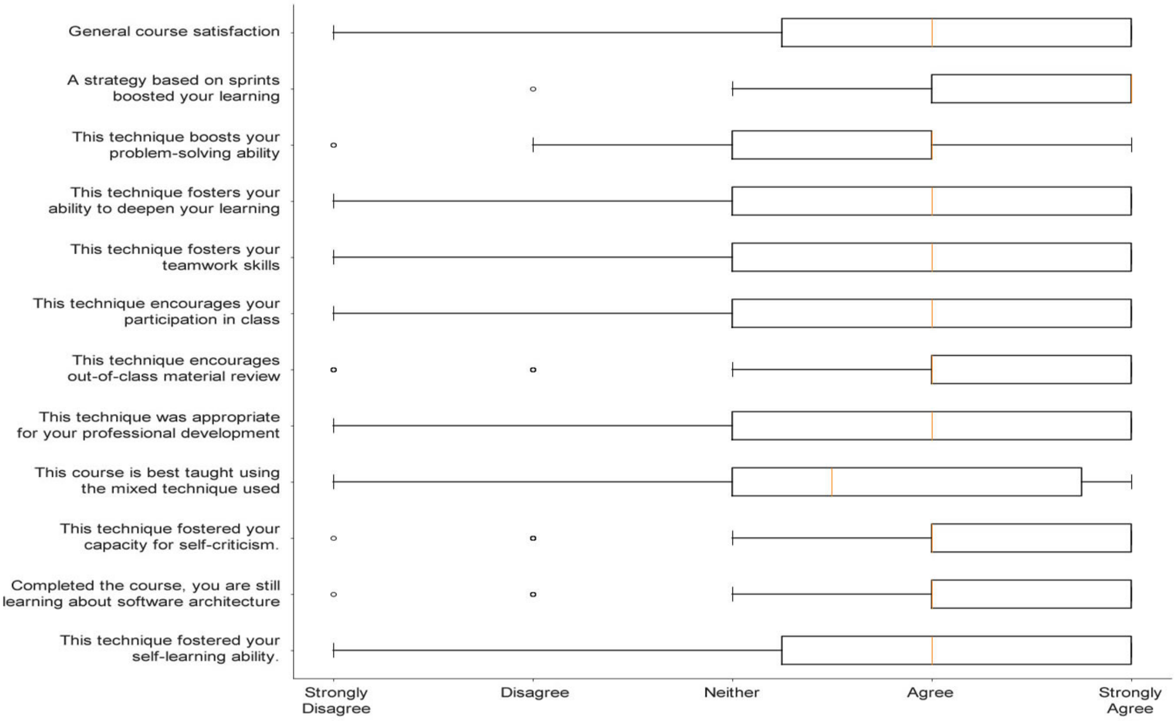


Figure 1. Post-course level of agreement with the blended learning methodology.

This survey was looking to understand if the methodology used was useful to improve skills such as self-criticism, self-learning, teamwork, problem-solving and professional development. Below are presented the results for these skills.

- Self-learning. Almost 75% (39.7% agree and 34.9% strongly agree) of the respondents were positive about this blended learning methodology supporting this skill. Only one (1) respondent was strongly disagree with this question, however, this same respondent stated that what he liked most about the course was precisely autonomous learning. This means the student was curious enough to explore the topics by himself. In that sense, the course did promote self-learning.

- Self-criticism. Almost 75% of the respondents were positive about this blended teaching technique supporting this skill. However, asking them what they liked most about the course, most of the responses were focused on the teaching technique, the content of the course and the industrial experience of the teacher. This means that the support of this skill was a benefit derived from the teaching technique that was not consciously present in students' minds.
- Teamwork. In this skill, 36.5% of the respondents were agree (23 out of 63) and 30.2% were strongly agree (19 out of 63). Teamwork is the collaborative effort of interdependent individuals working together to achieve a common goal. This skill requires dealing with others' knowledge, attitude, and personality.
- Problem-solving. Problem-solving requires effective and continuous communication with other peers. If this communication fails or becomes slow, there is a risk of losing the continuity of the problem to be solved and failing to find the optimal solution. Almost 50% (31 out of 63) of respondents were Agree that this blended teaching technique supports this skill. Disagree and strongly disagree were 6.3% (4 out of 63) and 3.2% (2 out of 63) respectively. Respondents in this group of negative perception criticized the small number of face-to-face classes of the course.
- Professional development. Analysis of this skill is important because we want to know if the knowledge taught in the class had an impact on their professional lives. In this question, respondents strongly agree were 34.9% (22 out of 63) and respondents agree were 31.7% (20 out of 63). Strongly disagree had 7.9% of respondents (5 out of 63).

4.3. Students' perception through open-ended questions

Open-ended questions allow discovering responses spontaneously; avoiding the bias that may result from suggesting responses to the interviewee. However, open-ended questions require a coding system to organize the answers and allow their analysis.

In this survey, we used two open-ended perception questions (PQ): PQ1 (what do you think could be improved from the course?) and PQ2 (what didn't you like about the course?); the first question had 34 valid answers, and the second question had 30 valid answers. It is important to note that an answer could be codified into two or more categories because of its content.

4.3.1. PQ1 (what do you think could be improved from the course?). Answers where codified into seven categories such as: more industry case studies, more communication with the professor, more working material, more in-place classes, more homework, better organization of class time and others (Figure 2).

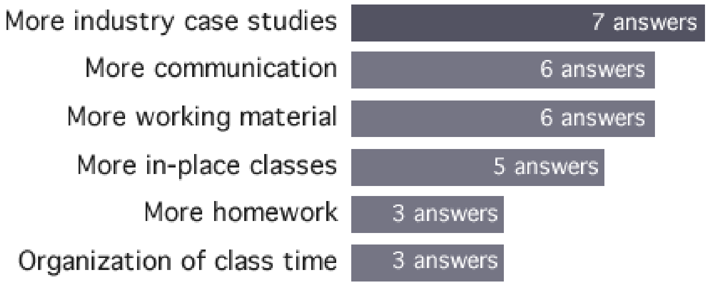


Figure 2. Codification of answers for PQ1.

Based on the categories identified, we could establish that students request more real industry case examples to be included in the course. Industry case studies allow students to understand the contexts under which architectural decisions were made. Sometimes, architects make sub-optimal decisions but under certain conditions, they can result in successful decisions. Moreover, it is clear that theory and practice do not always come together. Respondents also expressed their need for better and more fluid communication with the professor. The formulation of questions is part of the learning process, and many times, the answer to these questions is a fundamental piece in the consolidation of knowledge. Therefore, a quick answer or communication with the teacher will greatly benefit this learning process.

4.3.2. PQ2 (what didn't you like about the course?). Answers where codified into six categories such as: blended aspect, little interaction with the teacher, teacher's assistant failures, methodology, evaluation and others. Results of the codification can be seen in Figure 3.

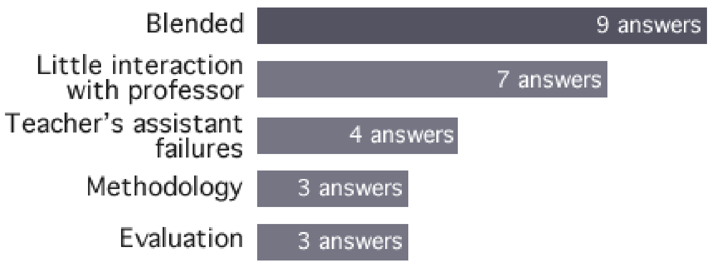


Figure 3. Codification of answers for PQ2.

The blended feature is a central aspect of our teaching technique. It is interesting to note that this was the most criticized aspect. Unfortunately, a lot of respondents didn't explain it, so the answer keeps at the half. This uncertainty creates an opportunity to deepen understand why students are uncomfortable with this aspect. "Teacher's assistant failures" was an unexpected category. We were expecting answers about the blended teaching technique and respondents expressed their disapproval of the work done by the teacher's assistant. This is evidence that students evaluate a course from all perspectives. In addition, this category reinforces the concept that open-ended questions allow the interviewer to find more than he/she anticipated.

5. Conclusions

This paper reported on the students' perceptions of learning outcomes of the agile software architectures course using a blended learning methodology. We measured students' perceptions by surveying all students who are taking the course from 2015 to the first semester of 2018. A total of 63 students out of 227 answered the survey.

Respondents agreed that this approach support self-learning and self-criticism. Self-criticism was a benefit derived from the teaching approach that was not consciously present in the students' minds. Teamwork and professional development were also agreed on by most respondents.

Students demand more real industry cases to be included in the course. This is an important request because the problems of software architecture are only visible at scale, and disappear once small projects as examples are used [1]. Respondents also demanded more and better communication with the professor. For this, the course included a forum to help students with any issue or problem they were facing. However, respondents said that this solution is not fast enough and sometimes it led to more questions.

In general terms, students agreed on the importance and satisfaction of using this blended teaching technique to learn about software architecture. However, its effectiveness could not be demonstrated. This could be caused by the few face-to-face classes of the course. Students stated that they preferred to be in constant interaction with the professor in order to clarify issues or have a better understanding of the concepts.

This study leaves interesting open questions that require further research. Particularly, what aspects of the blended aspect of our teaching technique can be improved? What techniques can be applied to improve communication between students and professors? How can self-learning be fostered in students with little knowledge of a particular subject? For future work, we encourage other researchers and professors to study how a combination of teaching techniques can improve the learning process in software architecture.

References

[1] Van Deursen A, Aniche M, Aué J, Slag R, De Jong M, Nederlof A and Bouwers E 2017 A collaborative approach to teaching software architecture *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17)* (New York: Association for Computing Machinery) pp 591–596

[2] Mannisto T, Savolainen J and Myllarniemi V 2008 Teaching software architecture design *Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008)* (Vancouver: IEEE) pp 117–124

[3] Lago P and Van Vliet H 2005 Teaching a course on software architecture *18th Conference on Software Engineering Education & Training (CSEET'05)* (Ottawa: IEEE) pp 35–42

[4] Urrego J S and Correal D 2013 Archinotes: A tool for assisting software architecture courses *26th International Conference on Software Engineering Education and Training (CSEE&T)* (San Francisco: IEEE) pp 80–88

[5] Galster M and Angelov S 2016 What makes teaching software architecture difficult? *IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)* (Austin: IEEE) pp 356–359

[6] Angelov S and de Beer P 2017 Designing and applying an approach to software architecting in agile projects in education *Journal of Systems and Software* **127** 78

[7] Mahnič V 2015 Scrum in software engineering courses: An outline of the literature *Global Journal of Engineering Education* **17(2)** 77

[8] Cleland-Huang J, Babar M A and Mirakhorli M 2014 An inverted classroom experience: Engaging students in architectural thinking for agile projects *Companion Proceedings of the 36th International Conference on Software Engineering (ICSE Companion 2014)* (New York: Association for Computing Machinery) pp 364–371

[9] Gehringer E F and Peddycord III B W 2013 The inverted-lecture model: A case study in computer architecture *Proceeding of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE '13)* (New York: Association for Computing Machinery) pp 489–494

[10] Mason G S, Shuman T R and Cook K E 2013 Comparing the effectiveness of an inverted classroom to a traditional classroom in an upper-division engineering course *IEEE Transactions on Education* **56(4)** 430

[11] Paasivaara M, Vodă D, Heikkilä V T, Vanhanen J and Lassenius C 2018 How does participating in a capstone project with industrial customers affect student attitudes? *IEEE/ACM 40th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)* (Gothenburg: IEEE) pp 49–57

[12] Mahnič V 2012 A capstone course on agile software development using scrum *IEEE Transactions on Education* **55(1)** 99

[13] Bastarrica M C, Perovich D and Samary M M 2017 What can students get from a software engineering capstone course? *IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)* (Buenos Aires: IEEE) pp 137–145

[14] Perez B, Castellanos C and Correal D 2018 Applying data mining techniques to predict student dropout: A case study *IEEE 1st Colombian Conference on Applications in Computational Intelligence (ColCACI)* (Medellín: IEEE) pp 1–6

[15] Easterbrook S, Singer J, Storey M A and Damian D 2008 Guide to advanced empirical software engineering *Guide to Advanced Empirical Software Engineering* (London: Springer) pp 285–311