Flipped Classroom Applied to Software Architecture Teaching

1st Anderson Cavalcante Gonçalves
Institute of Informatics
Federal University of Goiás
Goiânia, Brazil
andersoncavalcantegoncalves@inf.ufg.br

2nd Valdemar Vicente Graciano Neto
Institute of Informatics
Federal University of Goiás
Goiânia, Brazil
valdemarneto@ufg.br

3rd Deller James Ferreira
Institute of Informatics
Federal University of Goiás
Goiânia, Brazil
deller@inf.ufg.br

4th Uyara Ferreira Silva
Institute of Informatics
Federal University of Goiás
Goiânia, Brazil
uyara.silva@ifg.edu.br

***Abstract*—Teaching (and also learning) software architecture is often challenging since it requires a high degree of abstraction from the students besides talent and creativity of the instructors. Given that context, flipped classroom (FC) has emerged as an approach with potential to reinforce software architecture learning. FC learning promotes a greater immersion in a topic than traditional teaching, improving student engagement, interaction and cooperation by offering students the content before the classroom. Despite the FC popularity and its potential to enhance software architecture learning, studies on the use of it in software architecture teaching are still scarce. The main contribution of this paper is then presenting results on the adoption of FC applied to software architecture teaching. A case study involving a lesson on software architecture inside a course on Internet of Things was taught to a control group (19 people taught using traditional method) and to a treatment group (54 people taught using flipped classroom). Results reveal that the treatment group (who received the intervention via the FC), had a significant better (i) assimilation of content (73.1% of the students acquired most of the content according to the applied assessment), (ii) easiness of abstraction (81.2% of students declared that understood the essential concepts), and also (iii) satisfaction with the technique (78.07% of the students were satisfied about the approach). Conversely, the application of traditional method revealed (i) a lower degree of content assimilation (only 19.1% retained most of the applied content), (ii) only 39.9 % of the students understood the fundamental concepts, and (iii) only 25.87% of the students were satisfied and feeling confident about the content learned.**

***Index Terms*—Inverted Classroom, Software Architecture, Flipped Learning, Teaching**

I. Introduction

Software architecture refers to the structure of the software, its interfaces, and the interaction between all elements involved in the operation and setting of a software [1]. It is the set of structures necessary to understand the system, the elements, relationships between them, and their properties [2] and comprises the backbone of any well-succeeded software system [24]. Software architecture is essential in software analysis and design; it is a process that involves the creativity of the software architect to design software that meets the functional and nonfunctional requirements of a system [3]. Such features make software architecture an essentially abstract topic that is difficult to understand. The problem to be addressed in this paper involves the complexity of teaching software architecture and the proposal of a learning model that favors teaching and learning. Teaching software architecture is challenging, since (i) it requires proper analysis to fully understand its complexity and (ii) students need to practice theoretical and technical skills to become good software architects [4]. A poorly founded and structured architecture project can compromise the entire later structure of software organization and development, and may cause major harm to developers and institutions [6]. That is why teaching software architecture is essential for training architects with the right ability to design software.

The main contribution of this paper is to report results on the adoption of the flipped classroom (FC) approach to a software architecture lesson during an Internet of Things (IoT) course. In FC, the teacher/professor acts as a mediator, coordinating the questions and providing expert feedback on what is being addressed. The content of the classes is sent to the students before the classroom meeting, with access to the material in advance, the students can study the content in text and video format,which will allow the work to be done in the classroom. In this learning model the teacher will have the opportunity to mediate group discussions and help students solve specific problems. Other studies have already revealed that FC learning has shown great potential to improve the teaching and learning process [8]. Our results endorse those findings by also obtaining a significant better result on applying FC when compared to traditional classes to teach software architecture.

The remainder of this paper is structured as follows: Section II presents related work; Section III details the applied methodology; Section IV brings the report on the controlled experiment conducted; Section V presents a discussion on how FC is applied to teach software architecture in comparison to other disciplines in computing; and Section IV concludes the paper with final remarks.

II. Background

Software architecture involves the structure and organization by which modern systems components and subsystems interact with to form systems [9]. Teaching relationships and how system components and properties interact to form systems is difficult, since it requires significant abstraction from the students. According to Rupakheti and Chenoweth [5], teaching software architecture relates to the problem of making realistic learning. Most systems are too simple intrinsically to approach software architecture. An approach on how to teach architecture software at a high level was performed by analyzing prior approaches used over the years.

Naveda [10] describes a structure for an introductory software architecture course and the experience of teaching it. There is also an analysis of how the course can evolve over time. One of the topics studied is software design , ranging from the most concrete to the most abstract, such as subsystems and software architecture projects.

Professionals need to be able to address a variety of software quality issues that arise at different stages in the software development process. In traditional classes, students are expected to work on projects at home, after receiving formal laboratory classes. As didactic as the teacher is, teaching software architecture is a challenge, laboratory classes need to help students find solutions to the proposed problems. Over time, courses have evolved and involve solving problems in the laboratory, entrusting academic training to only a good book and a set of interesting laboratory classes is challenging [10].

Karam, Qian and Herrera [11] have developed a high level course, detailing the content and pedagogical approach to be followed in this course. All the details have been described for software architecture teachers and instructors to use as a tool to improve learning. They detail topics, strategies, and present an assessment of results, teaching experience, and a plan for further improvement. The results are satisfactory, however the teaching leaves gaps to be investigated.

Lieh and Irawan [12] remark that software architecture is an essential skill in software engineering, since such knowledge is essential for system design. They argue that architectures shape high-level visualizations and that developers can abrade unnecessary details and lose focus on the mainstream. Software education architecture requires addressing the problem of how to express the high level of abstraction and also making learning more realistic.

A study of years of software architecture teaching was carried out describing the main challenges encountered, analyzing approaches and tools to support students. According to Gaister and Angelov [13], they present a study that discusses what makes it difficult to teach software architecture and how software architecture differs from teaching other software engineering topics. Software architecture has specific characteristics that increase complexity and require a significant degree of abstraction and creativity.

An investigation was made on pedagogical approaches and tools that support the teaching of software architecture. The reported works investigate the main difficulties found in a teaching experience on software architecture domain and analyze pedagogical approaches and tools that could support it. In research, no studies propose the use of FC in software architecture teaching were found. The next topic will cover the applied methodology.

III. APPLIED METHODOLOGY

For teaching software architecture, we elaborated a controlled experiment study. The methodology was structured as follows:

Step 1. Content Preparation and Setting. For adopting FC approach, step 1 involved the prior preparation of content. The teacher prepared text and video content, which was made available through cloud storage service, a group was also created in the WhatsApp application, where the link for access to cloud storage was provided. A questionnaire was also designed to assess whether students had accessed and studied the available materials. Also through the group the teacher sent content-related questions to instigate group discussions.

Step 2. Classroom moment - controlled experiment execution. During the classroom, the teacher proposed activities for students to create their own definitions and micro software architecture projects to be applied in software design. Two groups were separately approached: one with the traditional method; and another one with FC. In the former, teacher acts as transmitter and the students as knowledge receivers as passive agents in the teaching and learning process [7]. In the latter, the student is invited to (previously to the class) watch a video and access the other content made available by the teacher. Then, during class, the student goes on assignments and group discussions, which can help to promote learning.

Step 3. Results analysis and reporting. This step comprised the analysis of results obtained by comparing the traditional method and FC to teach software architecture and reporting the results. After performing the proposed activities, the control group and the treatment group were submitted to assessment tests to measure the results. The questionnaire was used to collect the necessary data to investigate whether the objectives of the study were achieved.

The questionnaire consists of a sequence of logical steps, such as the planning of what will be measured, the formulation of the questions and the visual aspect [14]. Regarding the use of the questionnaire, it is observed that this instrument had a great potential to collect information about the application of the elements proposed in the paper. Those students involved in the survey answered several questions and, at the end, an analysis was made from the information collected [15]. A specific evaluation of the taught content was performed, and then an analysis of the averages obtained by the students was made.

The next section reports on the entire study.

IV. CONTROLLED EXPERIMENT EXECUTION

The controlled experiment was conducted to investigate whether FC has the potential to favour the teaching of software architecture. The objective of this study is to assess whether the use of FC learning applied to software architecture teaching promotes learning improvement in regards to the traditional method. Moreover, the research also seeks for understanding whether the proposed method can help the teacher on promoting interaction and learning during the process.

The approach related to the application of FC learning involves a control group and an experimental group. The former group faced the software architecture classroom by following a traditional teaching model, with an expository approach and the presentation of examples. The classes involved in the study are related to the disciplines of Programming and Internet of Things, with the theme of Apps Development. The content taught involved concepts related to software architecture and the importance of software architecture for software design. In turn, the latter group went under the application of the FC approach. The content taught in class was sent in advance in text and video format to students, with a questionnaire to be answered by them and interaction through social network. The use of the questionnaire and interaction via social networking application aimed to ensure that students would study the proposed contents. During the FC, a work mediated by the teacher to discuss the past contents in text and video was proposed. The work aimed to stimulate interaction and collaborative learning. The aim was to prompt students to engage in productive discussions where they could analyze and infer opinions related to the views of their classmates. After the process, an activity was prepared for both groups and an assessment about the content presented by the teacher was applied.

Students also had the opportunity to report the difficulties encountered during the completion of the proposed activity and the assessment performed. Students were also given the opportunity to ask questions and answer group questions.

A. Study Organization

Control Group. The control group involved students attending the Programming discipline. The class of 19 students participated in software architecture studies using the traditional teaching method, in which flipped learning was not used.

Classroom Activity Control Group. As shown in Figure 1, the teacher gave lectures in which, at the end of the presentations, the students had the opportunity to ask questions. After explaining the proposed content, the teacher applied activities to the class. The first activity consisted of the fixing exercises in which students were asked to formulate their own software architecture conceptualization, and the second activity consisted on the elaboration of a micro software architecture project for the creation of software for communication in computer networks. A questionnaire was also elaborated to investigate if the students had previous knowledge related to software architecture before the lectures. All activities were performed outside the classroom.

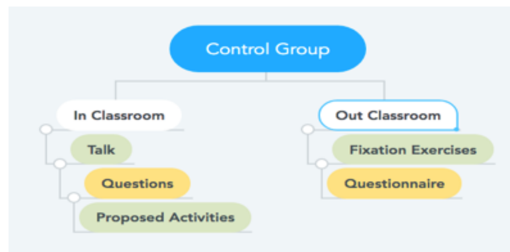

Fig. 1. Structure used in control group.

Performance Evaluation and difficulties reported by the Control Group. An evaluation with four questions was elaborated to assess if the students had understood the taught contents. Some difficulties were reported by the students involved in the control group.

Some difficulties were reported by the students, such as the high complexity and the difficulty of understanding about the contents related to software architecture and how such an area could be applied in software development practice. Next, w the study involving the experimental group will be addressed.

B. Experimental Group Study

The experimental group involved students from a class formed by the disciplines of internet in education and internet of things (IoT). The class worked together on the topic of IoT. Fifty-four (54) students participated in the experimental group. The class participated in software architecture studies using the FC model applied to software architecture teaching.

The flipped classroom model applied to software architecture teaching was structured as shown in Figure 2, in which we observe the activities performed outside the classroom and the activities performed in the classroom. The activities outside the consisted of making content available to students through the Box cloud storage, including an article on software architecture, followed by a video exemplifying the contents presented in the article, later sent additional material in audio and promoted group discussion, with questions sent by the professor.

Next, the professor sent a link to the online activity to be performed by the students, the activity aimed to stimulate access to the content available. The classroom activities consisted of a group discussion, which analyzed the content acquired by the students. The professor then proposed activities to be performed in the classroom, and mediated the collaborative resolution of the activities. After, the professor applied an assessment and a questionnaire about the content addressed and the learning model used.

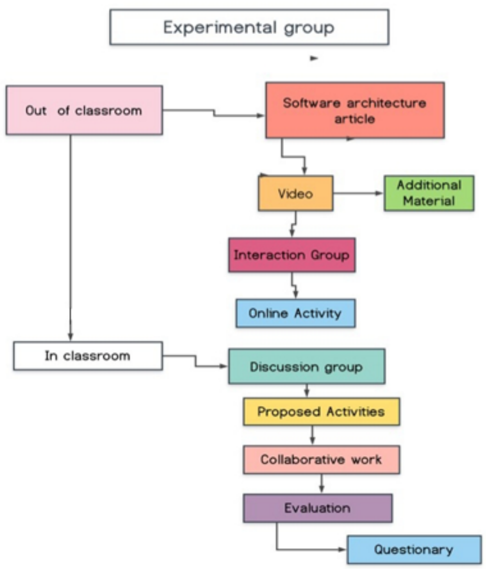

Fig. 2. Flipped Classroom Structure.

Activities Out Of Classroom. Activities outside the classroom were specified by the subject professor. A software architecture article, a video, and questions to answer in a group in the WhatsApp application were chosen and made available. To assess whether students accessed and studied the material provided, a link with specific questions to be answered by the students was also sent. The selected article contains important information about the concepts and introduction to software architecture, the video contains examples that help the student to understand the content proposed in the article, the online activity helps to fix the concepts learned by the student.

Materials made available outside the classroom were constantly available. According to Triantafyllou and Timcenko [16], out-of-classroom instruction and the use of online resources help to improve learning by providing detailed explanations that may involve students. The interaction of students in groups was analyzed by the discourse analysis method [17] used to assess the quality of online discussions.

Interaction in Questions and Answers. In the FC model applied to the teaching of software architecture, professor and students interacted through a social network group in which questions, doubts and questions were posted. The use of social networking enabled the teacher to help students and answer questions in the shortest possible time. Thus the teacher acted as a mediator of knowledge.

Teacher Mediation. In the FC model, the student is no longer a passive learning agent and actively participated in the teaching and learning process [18]. The professor acted as a mediator of knowledge, showing new ways, ideas, proposing challenges to be overcome. The teacher is no longer the sole holder of knowledge and helps to point the learning path for students.

C. Results

The results were analyzed according to the students and professor answers to the elaborated questionnaire. The questions proposed for the discussion group were based on the structured question structure designed to foster productive meetings [19]. For the analysis of the data obtained with the application of the questionnaires, guidelines on discourse analysis and critical thinking were followed [17].

An analysis was performed in the control group and experimental group to verify the students positioning in regards to the teaching model used.

Control Group Results. To analyze the knowledge of the class about the concepts of software architecture concepts acquired, a questionnaire was prepared for the students. The control group involved 19 students. The class was asked about their level of knowledge about the software architecture area. 29.4% of the students said they had no knowledge on software architecture. 29.4% said they had heard about the software architecture area but did not have specific knowledge on the subject. 35.3% said they had knowledge about the existence of software architecture, but did not know how it was applied in the software design.

The control group was asked whether the knowledge acquired in the lecture could help on their software development practice. 82.4% said they believed so, 17.6% said it helped a little. Regarding the possible difficulties encountered, it was analyzed how these difficulties could be minimized. Students were asked if the content on software architecture previously submitted and discussed could help on their understanding and facilitate learning during the classroom. 82.4% answered yes, for sure, and 11.8% answered yes, somewhat, as shown in Figure 3.

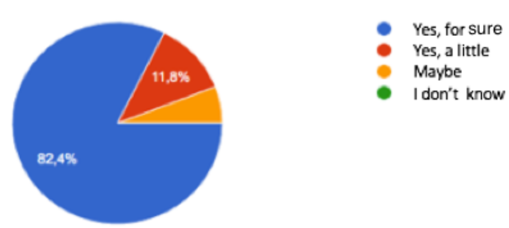

Fig. 3. Control Group Chart.

TABLE I
DISCOURSE ANALYSIS CONTROL GROUP.

Category	Average
Relevance	39.9%
Importance	27.3%
New Ideias and Solutions	19.1%
Critical Thinking	17.2%

TABLE II
DISCOURSE ANALYSIS EXPERIMENTAL GROUP.

Category	Average
Relevance	81.2%
Importance	77.4%
New Ideias and Solutions	73.1%
Critical Thinking	80.6%

The discourses produced by the students were also analyzed according to the discourse analysis and critical thinking guidelines [17]. As seen in Table I, the categories analyzed involved relevance, importance, new ideas and solutions, and critical thinking.

The analysis shows that 39.9% of the students presented relevant answers that addressed aspects related to software architecture. 27.3% presented important aspects for software architecture. 19.1% presented new ideas and solutions for the use of software architecture in software design. 17.2% of the students presented criticism related to the proposed examples that could contribute to the software development.

Experimental Group Results. To analyze the knowledge of the group involved in the experimental group about the concepts of software architecture that would be applied, a questionnaire was elaborated.

The class was asked about their level of knowledge about the software architecture area. 47.4% of students said they had no knowledge of software architecture. 5.3% said they had heard about the software architecture area, but had no specific knowledge on the subject. 47.4% said they had knowledge about the existence of software architecture, but did not know how it was applied in the design of software, a clear increase in regards to the control group.

The experimental group was asked if knowledge acquired outside the classroom helped on understanding software architecture. 89.5% said they believed so, 5.3% said they did not know and 5.3% said it did not help. It is observed that the absolute majority believe that knowledge acquired outside the classroom impacted on and supported them to understand the concepts studied.

Regarding the possible difficulties encountered, it was analyzed how these difficulties could be minimized. Students were asked whether software architecture content was constantly available for access, 100% of the students answered that the content was continuously available.

Students were also asked whether the access to the material used in the classes before the classroom assisted in the previous understanding of the topics covered, 94.4% answered yes, 5.6% answered that a little. This result reveals that early access outside the classroom helped in understanding the content related to software architecture.

Student Testing and Performance. An assessment with structured questions was designed in order to analyze student performance in the FC model applied to software architecture. Questions were also proposed to be answered collaboratively through social networking group. Students responses were analyzed according to the discourse analysis and critical thinking guidelines [17].

As seen in Table II, the categories analyzed involved relevance, importance, new ideas and solutions, and critical thinking.

The analysis shows that 81.2% of the students succeeded to present relevant answers addressing aspects related to software architecture. 77.4% presented important aspects for software architecture. 73.1% presented new ideas and solutions for the use of software architecture in software design. 80.6% of the students presented criticism related to the proposed examples that could contribute to the software development.

Student Perceptions on Teaching. A questionnaire was designed to ask students whether using Flipped Classroom improved student-professor interaction. As shown in Figure 4, to 89.9%, inverted classroom use improved student-teacher interaction, to 11.1% slightly, and no student responded that have not been improved.

We also asked if the professor acted as mediator, facilitator and was able encourage of the students. For 94.4% the professor acted as mediator, facilitator and under an encouragement engagement to the students. 5.6% answered that a little, and no student answered no. The research also sought to investigate whether classroom activities were more productive, even considering the period reserved to clarify doubts with the instructor. 90.7% answered yes, 9.3% answered a little and no student answered no.

Students were asked on a Likert scale from 1 up to 5 whether there was room for analysis of the topics addressed and whether the classroom activities fostered creativity and new ideas. One denoted the minimum intensity and 5 the maximum intensity about the issue being investigated. 59.3% responded with scale 5 representing maximum intensity, 24.1% answered with scale 4, 14.2% with scale 3 and 1.9% with scale 2. No student marked scale 1 as an answer.

The results reveal that the students perception regarding teaching was very satisfactory and productive.

Next, the students' perception of the flipped classroom will be analyzed.

Student perception on Flipped Classroom. The questionnaire also addressed issues related to students perception about FC approach. Students were asked whether the classes were more dynamic with the use of the that approach. For 94.4% of the students the classes were more dynamic, 5.6%

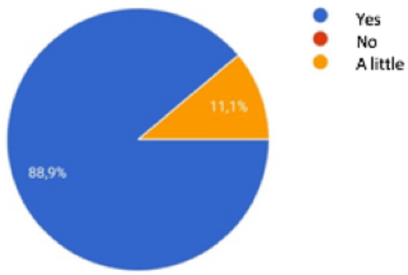

Fig. 4. Student Perceptions of Teaching.

said a little and no student answered no, as it can be seen in Figure 5.

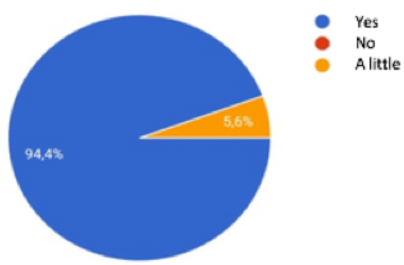

Fig. 5. Student Perceptions of Flipped Classroom.

They were asked whether the use of the FC contributed to the professor by offering individualized support to the students. 87% answered yes, 9.2% of students answered a little. Next, the professor perception of teaching and the use of the flipped classroom will be addressed.

Professor's Perception of Teaching. Questionnaires were prepared for the professor who was involved to the control group and to the experimental group to analyze the professor's perception of learning. The professor involved in the control group said he uses the traditional teaching method to promote student interaction in the classroom. Classes are expository and near the end of the class students have the opportunity to ask questions.

To promote student participation in class, the teacher proposes exercises and assignments to be performed outside the classroom. According to the professor, the collaboration of students in the classroom is stimulated by evaluative works in which grades are awarded. He said he believed it could be more productive for the classes if the content was previously studied at home by the students and that time in class was devoted to questions and discussions. However, the professor also said that a probable problem would be to ensure that students were engaged and motivate them to study the proposed materials outside the classroom.

According to the professor, during group work students shared tasks and each one was responsible for their own task.

TABLE III
DISCOURSE GLOBAL ANALYSIS.

Control Group	Experimental Group
25.87%	78.07%

The professor also claimed that he was content to acquire new teaching methodologies that could help in the teaching and learning processes.

Professor's Perception of the Flipped Classroom. According to the teacher who participated in the experimental group, the students dedicated themselves and performed the proposed activities as planned. The classes were more dynamic with the use of FC model, which contributed to the teaching and learning process. According to the teacher the preparation of classes for the FC was no more difficult than in other methods, but required dedication.

The construction of the content of the classes did not require more working time, but the classes had to be planned according to the proposed teaching model. The teacher stated that providing access to classroom material prior to the classroom helped students in classroom discussions and that the use of the method was very productive. Next, we will analyze the results obtained.

D. Analysis of Results

The results show that the FC helped significantly in teaching and learning, that students actively participated in the classes, elaborated more relevant, important answers, proposed new ideas and solutions and developed critical thinking, which contributed to learning. more solid. Students who participated in the experimental group achieved better results and proposed more efficient solutions to the problems presented by the teacher.

Regarding the concepts attributed in the proposed activities, the experimental group obtained better result, in the discourse analysis, the experimental group also obtained better result, as seen in Table III.

Table III shows the overall result obtained in the discourse analysis of the control group and the experimental group, the experimental group obtained 78.07% and the control group obtained 25.87%. This result shows that the experimental group is better able to analyze and propose solutions related to software architecture after the experiment. The next section establishes a discussion relied on a parallel between FC applied to software architecture teaching when compared with other FC applied to other disciplines in computing.

V. DISCUSSION

Other studies have reported results on the use of FC in other computing subareas [20], [21], [22]. The *main differences* between these studies and ours are (i) the use of FC specifically for teaching software architecture and (ii) the professor participation during the group interaction. In other studies, teachers adopt self-regulation before the classroom [23], [25], i.e., students need to self-regulate their own activities before the classroom. In our approach, the teacher mediates/regulates the process before the classroom and in the feedback of online activities.

Software architecture requires a higher level of abstraction from the student and from the practitioner. That characteristic imposes additional levels of complexity and difficulties to teaching and learning software architecture. In software architecture students face complex and poorly structured problems, having the ability to solve problems at this level is very valuable, students need to be able to propose practical solutions and act actively in solving problems [26]. In our approach, professor mediation helped students to acquire a complex knowledge that requires a high level of abstraction of content specific to software architecture, which sounds to be *another contribution of this study* regarded to the effects of using FC in that domain when compared to other domains in computing.

The traditional model of teacher-centered teaching has been refined by positive learning experiences based on active and collaborative, student-centered learning, which during learning engages in work aimed at analyzing appropriate solutions to improve teaching and learning, that generate a greater level of engagement and commitment to teaching, the acquisition of new knowledge and the skills to deal with complex problems [27]. Studies that use FC learning applied in computing provides the efficiency of a learning strategy focused on the student. However, there are complex challenges to be solved by the teacher, such as the student's involvement in the moment before the classroom, during the classroom and after the classroom, with the analysis and validation of the results. However, the use of self-regulation as a solution is reported and analyzed in some scientific research [28]. Studies also point out that students are often not prepared to self-regulate their activities in the FC [29]. The use of FC in the teaching of software architecture helped students to understand the concepts more easily, which enabled an improvement in the ability to analyze issues related to architecture.

Threats to Validity and Limitations

Empirical studies inherently suffer from threats to validity and limits to the potential of generalization of its findings. In this section, we discuss the main limitations on the application of FC approach found in the literature and shared with our study. Difficulties on applying FC can be ultimately converted on threats to the validity of the conclusions and findings obtained. We can mention some difficulties of applying FC. They are related to (i) difficulty in ensuring that students do the activities properly before the classroom; (ii) how to know if students are actually watching videos [30]; and (iii) to involve students and motivate them to participate in the proposed activities.

To relieve the difficulty (i), we conducted a group interaction for students to discuss the topic of the video and an online activity was also developed to assess whether students actually watched the video. In addition to the video, additional text material was also made available to help in the studies of the topic addressed.

There are also reports communicating that it is difficult to implement FC because access to technological resources in some places is very uneven [31]. However, researches have pointed out that, despite the limitations that can restrict the participant to an flipped classroom, students results show that the inversion is favorable [31], which relieves the threat related to difficulty (ii). To minimize the impact of this threat, we use interaction platforms that work on mobile devices, making it possible to carry out all activities through a smartphone, ensuring its presence during the activity.

Some difficulties may arise during the conduction of the FC, particularly at the moment in the classroom, related to difficulty (iii). The teacher must be prepared to resolve questions and eliminate doubts about learning. If these items are not observed, such limitations may threaten the success of the use of FC [32]. To reduce the risks associated with this, the teacher accompanied the students in the classroom, providing individualized assistance and seeking to resolve all students' doubts.

In this paper, the limitations were mitigated by the professor, the validation was centered on student performance during interactions, in which relevance, importance, proposition of new ideas and solutions and critical analysis of the content presented were analyzed. The results presented show that the use of FC in the teaching of software architecture was well-succeeded. Next section concludes the paper.

VI. FINAL REMARKS

This paper report results on a controlled experiment conducted to evaluate the impact of using flipped classroom (FC) approach for teaching software architecture when compared to traditional exposition methods. Students played an active role in learning by being involved in activities outside and within the classroom. The strategies used in the proposed teaching model helped to ensure that students accessed and studied the material before and after reaching the classroom. In the classroom, the teacher proposed assignments and exercises in which students obtained satisfactory results.

The results of the experimental group were superior to the control group in which there was no application of the teaching proposal of this article. There were more opportunities for students to opine and participate in group discussions and to analyze and critique solutions proposed by other students. However, it was observed that the application of this teaching model requires dedication and commitment from the teacher. Despite the excellent results obtained, one difficulty observed is related to the strategies that the teacher can use to promote group interaction. For future work, a teaching method that improves the application of the FC and that promotes student interaction and participation may be proposed.

Acknowledgments

Authors thank the students involved in the study. Authors also thank the State University of <Hidden> by providing the environment where this study was conducted and the Federal University of <Hidden> by supporting this study.

References

[1] Shaw, M and Garlan, D. "An Introduction to Software Ar- chitecture," V. Ambriola and G. Tortora, eds., Advances in Software Engineering and Knowledge Engineering, vol. 2, World Scientific Publishing, 1994, pp. 1–39

[2] Bass, Len., Clements, Paul., Kazman, Rick. Software Architecture in Practice. Addison-Wesley Professional, Pearson, 2013.

[3] Sommerville, I. Software Engineering. 10. ed. United States of America: Pearson Education, 2016.

[4] Van Deursen, Arie., Aniche, Maurício., Aué, Joop., Slag, Rogier., De Jong, Michael., Nederlof, Alex., Bouwers, Eric. A Collaborative Approach to Teaching Software Architecture. Delft University of Technology. ACM SIGCSE, Seattle, USA, 2017.

[5] Rupakheti, Chandan R., Chenoweth, Stephen. Teaching Software Architecture to Undergraduate Students: An Experience Report. IEEE/ACM 37th IEEE International Conference on Software Engineering, 2015.

[6] Pressman, Roger S. Software Engineering: A Practitioner's Approach. New York; United States: McGraw-Hill, 2009.

[7] Blanco, A., Nuñez, M., Gene, O. e Medina, J. (2017). Micro flip teaching - An innovative model to promote the active involvement of students. Computer in Human Behavior. Vol. 72. Pág 713-723.

[8] Lin, Y. (2019). Impacts of a flipped classroom with a smart learning diagnosis system on students' learning performance, perception and problem solving ability in a software engineering course. Computer in Human Behavior. Vol. 95. Page 187-196.

[9] Krutchen, P., Obbink, H., Stafford J. The Past, Present and Future of Software Architecture. IEEE Software. IEEE Computer Society, 2006.

[10] Naveda, J. F. (1999). Teaching architectural design in an undergraduate software engineering curriculum. Designing the Future of Science and Engineering Education.

[11] Karam, O., Qian, K., Diaz-Herrera, J. (2004). A model for SWE course "software architecture and design". Frontiers in Education, FIE.

[12] Lieh, O., Irawan,Y. (2018). Teaching adults learners on software architecture design skills. Computer and Systems Architecture Commons, Software Engineering Commons, and the Systems Architecture Commons.

[13] Galster, M., Angelov, S. (2016). What Makes Teaching Software Architecture Difficult?. Software Engineering Companion, ICSE-C.

[14] Aaker, E. Marketing Research. John Wiley and Sons, 7.ed, New York, 2001.

[15] Kitchenham, B.; Pfleeger, S. (2002). Principles of survey research: part 2: designing a survey. ACM SIGSOFT Software Engineering Notes, 27(1):44–45.

[16] Triantafyllou E., Timcenko O. (2015) Out of Classroom Instruction in the Flipped Classroom: The Tough Task of Engaging the Students. Learning and Collaboration Technologies. LCT 2015. Lecture Notes in Computer Science, vol 9192. Springer, China.

[17] Newman, D. R., Webb, B., e Cochrane, C. A content analysis method to measure critical thinking in face-to-face and computer supported group learning. St. Louis: University of Missouri–St. Louis, 1996.

[18] Yilmaz, R. (2017). Exploring the role of e-learning readiness on student satisfaction and motivation in flipped classroom. Computer in Human Behavior. Vol. 70. Pp 251-260.

[19] O'Donnel, A., and King, A. Cognitive Perspectives on Peer Learning. New Jersey: Lawrence Erlbaum Associates, 1999.

[20] Puarungroj, W. Inverting a Computer Programming Class with the Flipped Classroom. The Twelfth International Conference on eLearning for Knowledge-Based Society, p.40.1-40.7. Thailand, 2015.

[21] Alhazbi, S. Using Flipped Classroom Approach to Teach Computer Programming. IEEE International Conference on Teaching, Assessment, and Learning for Engineering, p.441-444. Thailand, 2016.

[22] Shyr W. J., Chen C. H. Designing a technology-enhanced flipped learning system to facilitate students' self-regulation and performance. Journal of Computer Assisted Learning. Wiley, 2017.

[23] Elakovich, D. M. Does a Student's Use of Self-Regulation Change in The Flipped Classroom?. Montana State University. Bozeman, Montana, 2018.

[24] Nakagawa E.Y., Oliveira Antonino P., Becker M. (2011) Reference Architecture and Product Line Architecture: A Subtle But Critical Difference. ECSA 2011. LNCS, vol 6903. Springer, Berlin.

[25] Shih, M., Liang, J. C., Tsai, C. C. Exploring the role of university students' online self-regulated learning in the flipped classroom: a structural equation model. Interactive Learning Environments, 2018.

[26] Ng, E. M . W., Integrating self-regulation principles with flipped classroom pedagogy for first year university students. Computers and Education, 2018.

[27] Zamorano, L. R. M., Sánchez, J. A. L., Caballero, A. L. G., How the flipped classroom affects knowledge, skills, and T engagement in higher education: Effects on students' satisfaction. Computers and Education, 2019.

[28] Lai, L. C., Hwang, G. J., A self-regulated flipped classroom approach to improving students' learning performance in a mathematics course. Computers and Education, 2016.

[29] Voronina, M. V., Moroz, O. N., Sudarikov, A. E., Rakhimzhanova, M. B., Muratbakeev, E. K,. Systematic Review and Results of the Experiment of a Flipped Learning Model for the Courses of Descriptive Geometry, Engineering and Computer Graphics, Computer Geometry. EURASIA Journal of Mathematics Science and Technology Education, 2017.

[30] Demirel, C. M. G. Exploring the Flipped Classroom: Possibilities and Limitations. University of Lisbon, Portugal, 2016.

[31] Rivera, V. M. Flipped Classrooms: Advantages and Disadvantages from the Perspective of a Practicing Art Teacher. State University of New York. New Paltz, New York, 2016.

[32] Aydin, B., Demirer, V. Flipping the Drawbacks of Flipped Classroom Effective Tools and Recommendations. Journal of Education and Instructional Studies in The World, p.33-40, 2016.