IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 10, 2019, accepted July 4, 2019, date of publication July 18, 2019, date of current version December 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2929507

Students' and Instructors' Perceptions of Five Different Active Learning Strategies Used to Teach Software Modeling

WILLIAMSON SILVA[1], IGOR STEINMACHER[2], AND TAYANA CONTE[1]
[1]Institute of Computing (IComp)—Federal University of Amazonas, Manaus 69080-900, Brazil
[2]School of Informatics, Computing and Cyber Systems—NAU, Flagstaff, AZ 86004, USA

Corresponding author: Williamson Silva (williamson.silva@icomp.ufam.edu.br)

This work was supported in part by CAPES through process 175956/2013; FAPEAM through Edital 009/2017; and CNPq through processes 311494/2017-0, 430642/2016-4, and 423149/2016-4.

ABSTRACT Unified modeling language (UML) is a modeling standard in the software industry. However, students face difficulties when learning how to model complete and correct UML diagrams, in part because the traditional teacher-centered approach still dominates in computer science. In an attempt to bridge this gap, and to improve the effectiveness of teaching and learning in computing courses, educators use strategies that encourage students' active participation in the learning process. The goal of this paper is to understand how active learning strategies influence the teaching and learning process of UML diagrams. We performed four case studies in undergraduate courses at two universities. We collected: (a) students' perceptions of each strategy through questionnaires and focus group sessions; and (b) instructors' perceptions of the challenges faced in using such strategies through semi-structured interviews. We found fifteen factors that influence the active learning of UML diagrams. Whereas the literature shows that students often positively respond to active learning strategies, our results show that some students have had negative responses to the new strategies. Additionally, we identified four challenges that instructors may face when employing active learning strategies. The contributions of this paper include: factors perceived by students about how active learning strategies influence the learning of UML diagrams, the challenges and difficulties faced by instructors in applying these strategies, and suggestions on how to circumvent challenges noted by instructors.

INDEX TERMS Active learning strategies, computer science education, empirical study, modeling education, software engineering education, unified modeling language.

I. INTRODUCTION

The capacity to abstract has become one of the most important skills in Computer Science courses [1], [2] since it enables an in-depth understanding of specific concepts or problems using different detail levels [1]. With abstract thinking, students and professionals can analyze and design models used during software development [1], [2].

Unified Modeling Language (UML) [3] has been adopted as a standard modeling language in the software industry for the graphical representation of design models [4] using diagrams. Despite this status, UML has been criticized for semantic inconsistencies, inadequate notations, and ambiguous diagrams and constructs [5], [6]. Instructors and researchers reported that students creating theses diagrams experience difficulties [7], [8] related to understanding the diagram syntax and semantics [8], organizing information on the diagrams [6], using associations of generalization–specialization type [6], [8]. These challenges can affect the final software quality since the diagrams may incompletely and incorrectly represent the software under development [8], [9].

These difficulties may be caused by the way that UML has been taught [9]–[11]. Most instructors conduct a teacher-centered learning environment to present theoretical concepts to students. The complex nature of Computer Science courses prevents instructors from easily finding a pedagogical approach that meets students' needs, resulting in an

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaobing Sun.

 This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

inefficient learning experience [12]. Thus, there is a gap between the abstract concepts taught in the classroom and the skills necessary for students to succeed in the industry [13].

Instructors need pedagogical strategies that provide students with a challenging environment and actively involves them in the learning process [14]. A promising approach that is gaining space in Computer Science is Active Learning (AL) [15], [16]. According to Bonwell and Eison [17], ''active learning'' is typically defined by educational researchers as learning that requires students to engage cognitively and meaningfully with the courseware, so they are ''involved with the information presented, really thinking about it (analyzing, synthesizing, evaluating) rather than just passively receiving it''. Active learning provides opportunities for learners to critically think about the content through a range of activities that help prepare learners for the challenges of professional situations [18]. In this paper, we use the term ''active learning strategies'' to refer to the vast collection of instructional or learning activities from which instructors can choose (e.g., reading, solving problems).

Extensive research has shown that active learning strategies are more effective than traditional lectures for promoting a wide range of desirable educational outcomes, including greater student learning and better retention in undergraduate science, technology, engineering, and mathematics (STEM) courses [16], [19], [20]. These strategies strongly link to modeling teaching, as they enable students to (a) elaborate multiple solutions to a given problem [21], [22]; (b) improve teamwork skills [22], [23]; (c) improve the oral and written communication skills used to model the diagrams [22], [23]. However, a few papers describe the use of these strategies in the teaching of software modeling. Teaching software modeling is important because it helps students to develop an abstract-thinking mentality and to comprehend the concept of designing the components (objects) of a system [24]. This paper aims to understand the influence of active learning strategies on teaching UML diagrams. We do not present the strategies themselves, but rather how students and instructors perceive their use, what challenges they face in the process, and the lessons learned. To guide our research, we designed the following research questions:

RQ1: How do undergraduate students perceive the active learning strategies employed during the teaching of UML diagrams?

RQ2: What challenges do instructors perceive when using active learning strategies while teaching UML diagrams?

To answer our research question RQ1, we qualitatively analyzed the data collected from questionnaires and focus groups [25] conducted in four classes from two different universities from Brazil: Federal University of Amazonas (UFAM), and State University of Maringá (UEM). This enabled us to better understand the difficulties and benefits perceived by the students when using these strategies. To answer RQ2, we conducted semi-structured follow-up interviews with instructors to investigate the challenges encountered when incorporating active learning strategies into the classroom. All data were analyzed using coding procedures [26].

The remainder of this paper is organized as follows: Section 2 presents the background and related works; Section 3 describes the case studies; Section 4 shows the qualitative results; Section 5 presents a discussion of the qualitative results; Section 6 shows the limitation; and, finally, Section 7 presents the final considerations and future works.

II. BACKGROUND AND RELATED WORK

This section provides a brief introduction to the background and related work.

A. TEACHING SOFTWARE MODELING

According to Agner and Lethbridge [27], modeling is a key activity in software development that has transformative potential if it is more widely and better used. A diagram is an abstract description of a problem that can be used to represent a possible solution [27]. UML diagrams are the main artifacts used to support brainstorming, analysis, and design, as well as to create system code [1]. UML is a diagrammatic notation that is widely used in the software industry as a language for specifying systems [28], [29]. For this reason, UML is used in academia as a tool for modeling systems and teaching object-oriented paradigms [28], [29].

However, teaching UML is a challenge for Software Engineering instructors due to the complexity of the modeling concepts [28], [29]. Besides, instructors face difficulties in using consistent teaching and learning strategies that align theory and practice so that students may have experienced similar to those of their future professional life [30]. The literature also reports that teaching specific content like software modeling through traditional lectures that use only theoretical lessons can be limiting and can fail to build students' professional skills [6]. This is partly because the lectures focus on acquiring technical knowledge through heavy workloads, rather than prioritizing active learning and integrating knowledge and skills more aligned with the professional reality [31]. To improve this scenario, the software engineering education community has dedicated a significant amount of effort to develop new pedagogic strategies that make teaching more attractive to students, mostly based on active learning strategies [27].

B. ACTIVE LEARNING STRATEGIES

To mitigate these problems, educational researchers suggest using active learning strategies aiming to provide students with new experiences and learning opportunities, improving students' overall learning [32]. A body of literature evidences the advantages of using active learning in the undergraduate curriculum in courses such as Mechanical Engineering [32]–[34], Engineering Systems Design [35], [36], Mathematics [37], [38], and Physics [39], [40]. Several studies show that active learning strategies are often more effective than traditional lectures to promote a wide range of desirable educational outcomes, including greater student learning and better retention in STEM programs [16], [19], [20].

Active learning is founded in social constructivism: students build knowledge through interaction with ideas, concepts, materials, and other artifacts [41], [42]. Teaching using active learning strategies require a change in how students acquire knowledge and conceptual understanding, countering passive learning. Junhua [41] and Briggs [15] argue that the constructivist view developed in active learning strategies aims to create teaching experiences that: (1) help students build a deeper understanding of theoretical concepts in connection with practical experiences; (2) facilitate the development of student skills; and (3) develop students' capabilities and dispositions to engage in collaborative project-based research and critical thinking.

Students assume a central role and take responsibility for their own learning by employing active learning strategies during classes – they must discover some of the knowledge necessary to carry out the work [43]. The instructor, acting as a facilitator, facilitates learning and works as a guide in this process. Students genuinely invest in learning and developing skills such as responsibility, teamwork, and the ability to synthesize learned concepts [16].

C. RELATED WORK

Instructors can use active learning strategies to engage students in classroom activities by making them active stakeholders of the learning process. Below we report some works in which the authors used active learning strategies to teach software modeling to undergraduate students.

García-Holgado et al. [44] implemented an active learning strategy based on projects to increase the success rate of software engineering students. Students worked on developing an analysis model (use case model and problem domain models) of a specific topic. The authors noticed an increase in the student success rate from 41.71% to 63.89%, reporting that, although they only applied it in a specific context, the experience can be adapted to similar contents in other degrees and universities.

Scanniello and Erra [45] used a strategy based on Think-Pair-Square while creating use case diagrams. In this strategy, the students started with individually modeling (Think), then worked in pairs (Pair), and finally worked in a group with an even number of students (Square) of four or more. The authors realized that use case diagrams improved as students moved from one stage to another of the process, and students improved their ability to work collaboratively.

Fioravanti et al. [46] reported their experience applying a Project-Based Learning (PBL) strategy combined with project management to create an environment in which students deal with managers and other stakeholders. The goal is to bring students closer to real-life experiences by developing a software project in the context of business modeling. The authors reported that including PBL in the classroom brings innovation and dynamism in modeling teaching. Unlike traditional lectures, PBL instigates students to take a proactive role and to be responsible for acquiring new knowledge. In general, students were enthusiastic and had positive perceptions about PBL and the importance of using real problems.

Similarly, Silva et al. [47] carried out an empirical study that investigated the influence of two active learning strategies: Problem-based Learning, and Learning from Erroneous Examples, in the teaching of UML diagrams. The authors concluded that the diagrams created using the two strategies presented similar levels of correctness and completeness. The qualitative results showed that both strategies helped in understanding the concepts of diagrams and improved the interaction among the team. However, the team size made the learning of the diagrams difficult.

The studies presented in this section focus not on the strategies used but on how instructors and students perceive these strategies during teaching processes. However, we noticed that only the last work presents qualitative results regarding the perception of the students on the strategies, reporting the benefits and drawbacks. Also, to the best of our knowledge, no studies analyzed the instructors' challenges and lessons learned when employing active learning strategies during software modeling teaching. It is worth investigating these challenges so that instructors can understand the use of these strategies from students' point of view as well as other instructors.

III. RESEARCH METHOD

The main goal of this study is to understand the influence of active learning strategies in learning UML diagrams. We conducted a qualitative study to analyze instructors' and students' perceptions about these strategies. By understanding the factors prevent or help students achieve learning goals when employing active learning strategies, we can help instructors understand how these strategies can be used in the context of software modeling education. To achieve our goal, we followed the method presented in Fig. 1 and explained in this section.

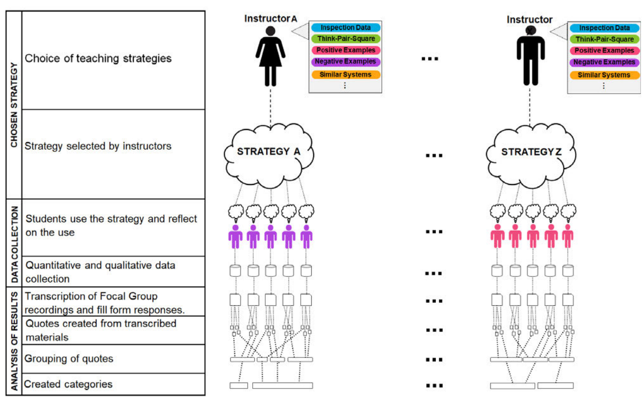

FIGURE 1. Research method overview.

A. CHOICE OF TEACHING STRATEGIES

To support instructors choosing the appropriate teaching strategy, we considered the strategies identified and curated

TABLE 1. Information about the class.

University	Instructor ID	Major	Course Name	Chosen Strategy	UML Diagram	# students
UFAM (Class A)	I01	Information Systems	Software Analysis and Design	Inspection-Based strategy Think-Pair-Square	Class diagram Sequence diagram	16
UFAM (Class B)	I02	Computer Science	Introduction to Software Engineering	Positive Examples Negative Examples Learning based on similar systems	Use case diagram Use case description Diagram and textual description	28
UEM (Class C)	I03	Computer Science	Software Engineering	Negative Examples	Class diagram	35
UFAM (Class D)	I04	Computer Science	Software Analysis and Design	Inspection-Based strategy	Sequence Diagram	21

in a Systematic Literature Mapping (SM) that we conducted previously [48]. From this SM, we identified technologies that have leverage any active learning strategy in their theoretical background. The learning strategies used were curated and catalogued in a web portal called OpenSMALS.[1]

(**Open** Repository for **S**oftware **M**odeling Teaching from **A**ctive **L**earning **S**trategies). The portal aims to assist instructors in choosing one or more strategies to use in the classroom, providing specific guidelines on how instructors can apply these strategies, modeling scenarios provided by other instructors, assessment questionnaires, among others. In addition, the tool offers a courseware repository that aims to help instructors apply the strategies, by offering, for example, modeling scenarios provided by other instructors or evaluation questionnaires. The web portal contains all the information about all strategies implemented in the courses by instructors.

At the initial stage of our research, instructors used this tool to select the strategy they would apply in their courses. The strategies selected by each instructor are presented in Table 1. Although it is part of the strategy, in this paper we do not discuss the choice of strategy.

B. PARTICIPANTS

The study was carried out in two Brazilian universities: State University of Maringá (UEM) and Federal University of Amazonas (UFAM). The study involved four instructors and 100 students. Table 1 presents an overview of the participants of this study.

Regarding Class A and Class B, the lectures took place in the first semester of 2018 at UFAM in Manaus-AM, Brazil. Both courses are part of the 5th term (senior year) of the curriculum and are the students' first contact with software modeling in the major. In these lectures, the instructors presented the concepts about Use Cases (UCs) modeling: diagram and textual description. After that, the students received an assignment to exercise a little more about the concepts taught. Class C and Class D lectures happened in the second semester of 2018. In Class C (UEM), students also had their first contact with software modeling in the Software Engineering course. In Class D, students had already learned software modeling and applied the concepts learned in a practical assignment in a previous course. These lectures had focused in: (a) presenting the fundamentals of software modeling, as well as the concepts of object-oriented analysis and design; (b) practicing the concepts using UML. To identify the students, we created an identifier, P##C, in which ## is a unique number for a student, and C represents the class they belong to. For example, P01A refers to student 01 from Class A. All the instructors have more than 12 years of experience in the software industry, and they have more than seven years of experience in teaching software modeling. Regarding the active learning strategies, all the instructors employed active learning strategies in the class, for example, Problem-Based Learning, Project-Based Learning, Flipped Classroom, and others. However, the strategies chosen by the instructors were new to them at the time of the study.

It is important to emphasize that the students voluntarily provided the data for our research. When instructors provided the consent form approved by the research board committee (process 2.545.694), the students were told that they could opt-out at any point, but they should carry out the assignment, which was part of the course grading schedule.

C. DATA COLLECTION

In order to compare the strategies, we applied post-modeling questionnaires aiming to evaluate the students' self-perceived learning. The students provided their answers in a 5-point Likert Scale, with options ranging from "I Strongly Disagree" (−2) to "I Strongly Agree" (2), with a neutral option.

The questionnaire items, presented in Table 2, were based on the dimensions of learning [30], [49]. Items 01 and 02 aimed at evaluating whether the strategies contributed to course learning outcomes; we designed items 03 to 08 according to Bloom's Taxonomy learning levels [50]; and items 09 to 18 aimed to assess the overall positive attitude toward the strategies [51], [52].

After answering the questionnaire, we invited all students to participate in focus group sessions [25]. All the students volunteered. Focus group is a technique for collecting qualitative data through group interviews [25]. The focus group configuration enables investigating on a specific topic, mediated by a moderator, in which the participants provide their

[1] https://sites.google.com/site/activelearningmethods/

TABLE 2. Items evaluated in post-model questionnaires.

	Item	Description of the Item
Factual Dimension of Bloom's Taxonomy	Item 01	The strategy contributed to my learning of the course.
	Item 02	The strategy was efficient for my learning, in comparison with other activities of the course.
	Item 03	The strategy helped me remember the concepts learned about the diagram.
	Item 04	The strategy contributed to understanding how the concepts learned can be useful to model the diagram.
	Item 05	The strategy contributed in applying the concepts of the diagram during problem resolution.
	Item 06	The strategy contributed to organizing the diagram during modeling.
	Item 07	The strategy contributed to verifying whether the diagram was correctly modeled.
	Item 08	The strategy contributed to create the diagram during modeling.
Attitudes of students about the active learning strategies	Item 09	With this strategy, I was more motivated to learn than usual.
	Item 10	The strategy enabled me to improve my critical opinions.
	Item 11	I feel more connected to others that employed this strategy.
	Item 12	The strategy made me feel part of the class.
	Item 13	With this strategy, I could express my opinions freely.
	Item 14	With this strategy, I discovered faults in what I had previously believed to be right.
	Item 15	I found this strategy very useful.
	Item 16	The strategy facilitates active learning.
	Item 17	The strategy supported me to examine issues and discuss in an argumentative format.
	Item 18	The strategy provides a good learning experience.

answers and discuss others' ideas, enriching the information obtained [25].

We split the focus group sessions into three steps. In the first step, we used a dynamic of lovers X haters [53] to encourage students to present their negative views. In this dynamic, we group volunteer students into teams, and each team has a predefined role during the discussion (see Fig. 2), in which lovers should argue in favor of the strategies, and the haters against them. At the beginning of the session, we defined the roles of each team (lover or hater) by lottery.

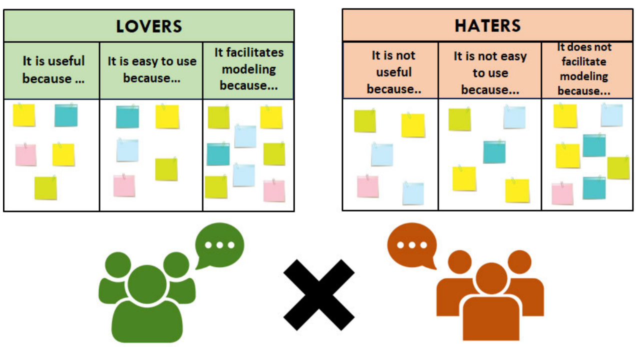

FIGURE 2. Board designed to guide the discussion of the Lovers X Haters Focus Group.

After this, (second step) each student used colored sticky notes to write the difficulties, positives and negatives aspects, or questions that they had when creating the model. Next, (third step) each student from each team placed their notes on a whiteboard (see Fig. 3) and explained them. The moderator asked the other students to comment on whether or not they agreed with the colleague's opinion. Each session lasted 1h30m, and all information from this session was video recorded by the moderator, with the consent of the participants who were present.

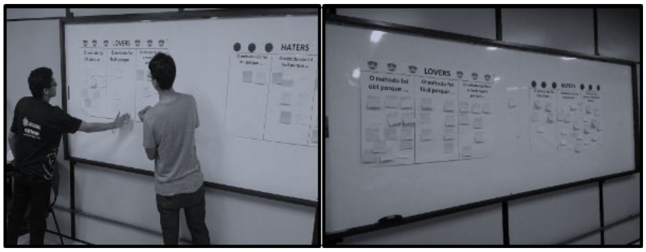

FIGURE 3. Focus group session.

After the instructors employed the strategies, we conducted semi-structured interviews, which consisted of a mixture of open-ended and specific questions designed to elicit anticipated and unexpected information types [54]. In this kind of interview, the questions are planned, and we seek to answer them, but they are not necessarily asked in the same way or the same order as they are listed [55]. We designed our interview script according to the literature recommendations [54], [55]. To meet the ethical requirements, the research goal was explained to each instructor, and a consent form was applied, guaranteeing the confidentiality of the data provided and the anonymity of the instructor. The interviews were performed individually, within the expected time (between 15 and 30 minutes). The researchers did not rush the interview and the interviewee was able to express herself/himself in a calm and uninterrupted manner. During the interviews, the instructor could express their perceptions about the use of the strategies during modeling teaching, as well as the challenges faced in applying these strategies.

D. DATA ANALYSIS

Our analysis was exploratory, aiming to generate new insights and provide us with a better understanding of the problem. We qualitatively analyzed the transcription of the answers following coding procedures [26]. The goal of the qualitative analysis was to code, categorize, and synthesize data, towards to identify the difficulties and benefits perceived by the students and instructors after using the strategies. Initially, we verbatim-transcribed all audio from the focus groups. We used Atlas.ti to support data analysis and synthesis. Data analysis began with open coding of the transcripts. Thus, we created codes (relevant concepts for understanding the perception about the learning strategies) related to students' answers (participants' quotes). We created post-formed codes as the coding progressed and attached them to particular pieces of the text. After this, we analyzed and grouped the codes according to their properties, forming concepts that represent categories (Fig. 4). The analysis was conducted by the first author and discussed with the other authors in multiple meetings. We followed this process to mitigate any potential bias in the coding process. The following

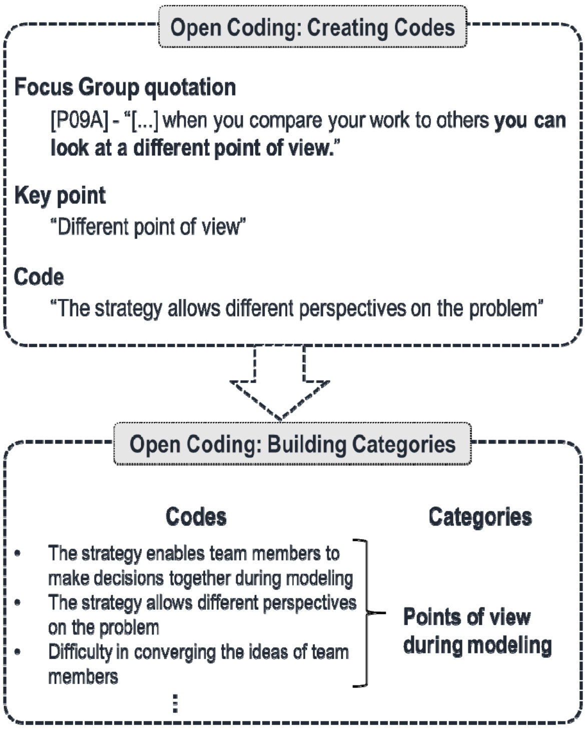

FIGURE 4. Coding example.

subsections present more detail about the qualitative results.

IV. RESULTS

In this section, we present our results according to the research questions.

A. RQ1: HOW DO UNDERGRADUATE STUDENTS PERCEIVE THE ACTIVE LEARNING STRATEGIES EMPLOYED DURING THE TEACHING OF UML DIAGRAMS?

To answer this RQ1, we first analyzed students' responses to our questionnaire about the perceptions of the students on the strategies employed. After that, we analyzed the students' comments during the focus group session and identified factors that positively or negatively influenced the use of the strategies.

1) STUDENTS' PERCEPTIONS ABOUT THE STRATEGIES

Figure 5 presents the perceptions of the students about their learning experience when using active learning strategies. We grouped the perceptions of the students about each of the strategies. Regarding the ''factual dimension'' of Bloom's taxonomy (items 3 to 8), most of the students agreed that the learning strategies positively contributed to learning. For the item that assessed whether the learning strategies contributed to learning (item 01), we obtained a good agreement level for 4 out of 5 strategies (91.7% agreement for Inspection-Based strategy, 81.1% agreement for Think-Pair-Square, 76.2% agreement for Negative Examples, 60.70% agreement for Positive Examples). The only contrasting result was for Similar Systems, for which only 46.4% of the students agreed. The Similar Systems strategy obtained the worst performance among all the strategies for the item that evaluates whether the learning strategies help in correctly modeling a diagram (item 07), with 21.4% disagreement. This discordance may relate to the fact that the learning strategy must have been used only to create diagrams. Overall, the results show that the use of these learning strategies promoted a good perception for the students, especially in terms of remembering, interpreting, and applying concepts.

Concerning attitudes, the Inspection-Based strategy evidenced the highest agreement levels. We noticed that more than 90% of the participants agreed that the strategy was useful for learning (item 15) and that they (students) participated more actively (item 16). However, we noticed that in item 09 (motivation to learn), a large number of students remained neutral (44.4%). Despite this, according to students' perceptions, these strategies contributed to the learning of diagrams. Regarding the Similar Systems strategy, we noticed a large number of participants disagreeing or remaining neutral when asked whether the strategy motivated them to learn (item 9); and if they (the students) felt more connected to their peers (item 11). Besides, more than 50% of the students remained neutral because they did not have a good experience in learning to model using this strategy (item 18). This shows that the Similar Systems strategy did not stand out as more efficient for creating diagrams.

2) QUALITATIVE RESULTS ON THE PERCEPTION OF STUDENTS ABOUT THE STRATEGIES

After analyzing the students' perceptions, we qualitatively analyzed their comments during the focus group session. We found five main factors identified in the qualitative study on active learning strategies (see Table 3).

a: POINTS OF VIEW DURING MODELING (FIRST FACTOR)

When we analyzed the data, we noticed that the strategies directly influenced the students' perspectives during the modeling of the solution. This factor relates to students' capacities to work in teams, in which there are different perspectives and in which team members express their point of view about the problem during modeling. This factor comprises benefits (a and b) and difficulties (c) perceived by the students (see Table 4): (a) the strategy enables team members to make decisions together during modeling; (b) the strategy allows different perspectives on the problem, and (c) difficulty in converging the ideas of team members. We highlight these subfactors when presenting the specific results in the tables.

b: COLLABORATIVE LEARNING (SECOND FACTOR)

The strategies assisted students who had difficulties in modeling to learn from other more experienced students, improving the quality of the collaboration, and encouraging students' communication and active contribution. Under this

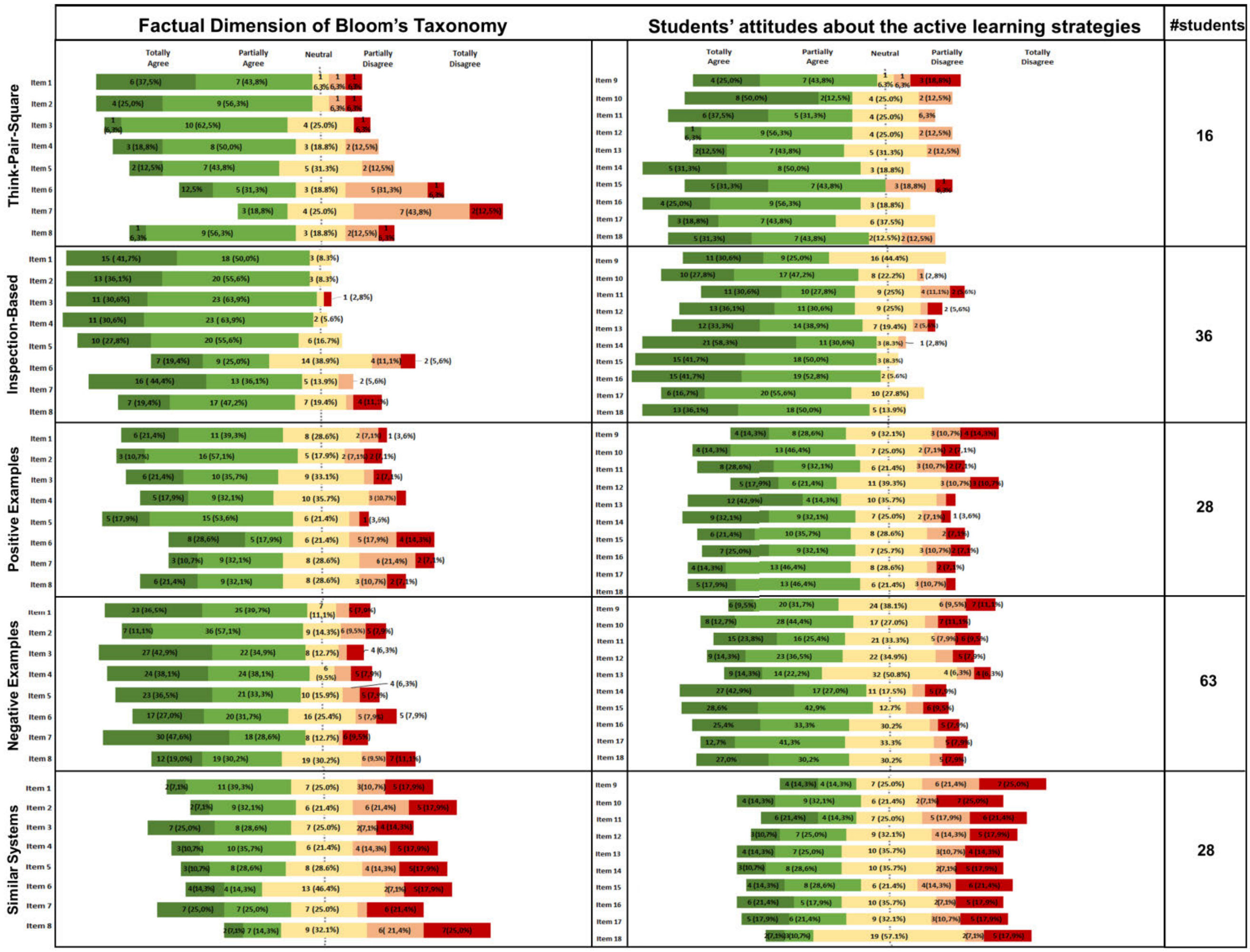

FIGURE 5. Students' perceptions about the strategies.

TABLE 3. Categories and subcategories identified in the qualitative study.

Factors	Subfactors	Active Learning Strategies: Think-Pair Square	Inspection Based strategy	Positive Examples	Negative Examples	Similar Systems
Influence from the student's point of view during modeling	(+) The strategy enables team members to make decisions together during modeling	X				
	(+) The strategy allows different perspectives on the problem	X	X			
	(-) Difficulty in converging the ideas of team members	X	X			
Collaborative learning	(+) The collaboration made it possible to create a diagram in more detail	X				
	(+) The interaction with the team allowed the learning	X	X			
	(+) The strategies provided the opportunity to learn how to work in teams		X	X		
Support during modeling	(+) The strategy guides students during modeling			X	X	
	(+) The strategy supports students in identifying defects during modeling			X	X	
	(-) The strategy does not support students during modeling			X	X	
	(-) Lack of moderator in decision-making groups	X	X			
The strategy allows the student to receive feedback	(+) The strategy makes it possible to receive feedback from the problems identified by the other team		X			
	(-) There was a lack of communication/interaction between teams (modelers and inspectors)		X			
Abstraction and problem solving	(+) The strategy makes it possible to better abstract the problem					X
	(+) The strategy integrates theory with practice					X
	(-) The strategy does not allow to abstract new scenarios					X

Note:
(+) It is regarding the benefits perceived by the students after using the active learning strategies;
(-) It is regarding the difficulties perceived by the students after using the active learning strategies.

category, we identified three benefits of using active learning strategies (see Table 5): (a) the collaboration made it possible to create a more detailed diagram; (b) the interaction with the team improved learning; and (c) the strategies provided the opportunity to learn how to work in teams.

TABLE 4. First factor: Points of view during modeling.

The strategy enables team members to make decisions together during modeling (evidenced by 2 students)
Think-Pair-Square: *"[...] first, we took all the points from all the diagrams and talked, then we got the best ideas [...] and (during modeling) people worked to put the ideas of everyone in the final version of the diagram."* – Student P17A
The strategy allows different perspectives on the problem (evidenced by 12 students)
Think-Pair-Square (3 out of 12): "*I thought the strategy was very useful because at each stage it seems that we are doing a more complete diagram [...] at each step we improved, in the end, we make a more complete diagram.*" Student P10A
Inspection-Based strategy (8 out of 12): "*[...] as we create our models and the other team gives feedback, we do not create a model based on our own opinion.*" – Student P03D
Difficulty in converging the ideas of group members (evidenced by 8 students)
Think-Pair-Square (7 out of 8): "*I had too many diagrams to compare, it confused me and brought more questions than it helped me.*" – Student P01A
Inspection-Based strategy (1 out of 8): *"we have spent a lot of time trying to reconcile different opinions within the team itself"* – Student P08D

TABLE 5. Second factor: Collaborative learning.

The collaboration made it possible to create a more detailed diagram (evidenced by 4 students)
Think-Pair-Square: "*the collaboration was great, because I was able to make a more detailed diagram.*" – Student P04A
The interaction with the team improved learning (evidenced by 8 students)
Think-Pair-Square (4 out of 8): "*even though I had missed the previous classes, I was able to assimilate the content and the idea, and I was able to participate during the modeling.*" Student P05A
Inspection-Based strategy (5 out of 8): "*as I had not created the diagram in the last class, I was kind of lost. When I started modeling with the team, the classmates were explaining to me and I learned quickly and I even managed to help.*" – Student P16D
The strategies provided the opportunity to learn how to work in teams (evidenced by 7 students)
Positive Examples (5 out of 7): "*[the strategy helped] further learning because of the opportunity to work in teams and to hear different opinions.*" – Student P08B
Inspection-Based strategy (3 out of 7): "*[...] there are different ways of working, but working as a team is just learning to give in and listen to the opinion of the others. I was thinking about getting the job done differently and my colleague had already decided about another way to work, but then we got it right.*" – Student P11A

c: SUPPORT DURING MODELING (THIRD FACTOR)

Students also reported positive and negative points related to the support provided by some of the strategies during modeling (see Table 6). This category included four factors: (a) the strategy guides students during modeling; (b) the strategy supports students in identifying defects during modeling; (c) the strategy does not support students during modeling; and (d) in the strategy, there is a lack of moderator in the decision-making groups. From these factors, a and b are benefits, while c and d are difficulties perceived by students.

TABLE 6. Third factor: Support during modeling.

The strategies guide the students during modeling (evidenced by 18 students)
Positive Examples (15 out of 18): "*when we did not know what to do in the discussion, we looked at the positive example worksheet (...) We relied on that to see if it was right." – Student* P17B
Negative Examples (3 out of 18): "*it helped to better understand how the use-case specification works.*" – Student P09C
The strategies supported the students in identifying defects during modeling (evidenced by 8 students)
Inspection-Based strategy (3 out of 8): "*[the strategy helped] solving the problem on time, and to avoid making silly mistakes and having to redo the diagram.*" – Student P10D
Positive Examples (1 out of 8): "*I was able to identify mistakes I made, so I could fix them.*" – Student P21B
Negative Examples (6 out of 8): *"[it was] very positive because it helped me to find my own mistakes that I was writing."* – Student P35B
The strategy does not support students during modeling (evidenced by 15 students)
Positive Examples (6 out of 15): *"[this strategy] contributed little, as most of the team neither looked at (the positive examples) and focused more on the activity itself."* – Student P10B
Negative Examples (9 out of 15): *"I already had questions and the negative examples did not help me, because I did not know to distinguish what was right from what was wrong (during the modeling)."* – Student P22B
In the strategy, there is a lack of moderator in the decision-making groups (evidenced by 6 students)
Think-Pair Square (3 out of 6) and Inspection-Based strategy (4 out of 6): "*if I ask a person who has the same level of knowledge as me, we'll have questions; but if I ask someone who actually knows how to model, then they will help me.*" - P03A

d: THE STRATEGY ALLOWS THE STUDENT TO RECEIVE FEEDBACK (FOURTH FACTOR)

From the results, we realized that the feedback provided was considered essential by students to improve modeling and can be a significant factor in students' motivation. We identified positive and negative aspects about feedback (see Table 7), which relate to the Inspection-Based strategy, since, among the strategies employed by the instructors, only this one has an explicit feedback step. We found that (a) the strategy makes it possible to receive feedback from the problems identified by the other teams, and (b) there was a lack of communication/interaction between teams (modelers and inspectors).

e: ABSTRACTION AND PROBLEM SOLVING (FIFTH FACTOR)

When we analyzed the data, we noticed that the Similar Systems strategy influenced students in abstraction and problem-solving (see Table 8). This category included three factors: (a) the strategy makes it possible to better abstract the problem; (b) the strategy integrates theory with practice; and (c) the strategy does not allow abstracting new scenarios.

TABLE 7. Fourth factor: The strategy allows the student to receive feedback.

The strategy makes it possible to receive feedback from the problems identified by the other team (evidenced by 5 students)
Inspection-Based strategy: "*when others correct our modeling, they identify errors that we could not see right away [...] and probably we would only see more up to the front when we realize that a certain thing is missing in some part of the project.*" – Student P10D
There was a lack of communication/interaction between teams (modelers and inspectors) (evidenced by 10 students)
Inspection-Based strategy: "*there is people who did not understand, the person reported the problem and we lost time trying to understand why she said that.*" – Student P08A

TABLE 8. Fifth factor: Abstraction and problem solving.

The strategy makes it possible to better abstract the problems (evidenced by 11 students)
Similar Systems: "*this strategy facilitated modeling because when you look at a problem that actually exists, it comes out a little bit from that abstraction and simulation that we have in the classroom.*" – Student P09C
The strategy integrated theory and practice (evidenced by 7 students)
Similar Systems: *"this strategy was one of the most useful because, in my opinion, it was the best way for us to apply the knowledge that we saw in the classroom in a practical way and in a real system."* – Student P22C
The strategy did not allow abstraction of new scenarios (evidenced by 3 students)
Similar Systems: *"it was very hard using a ready system because I had to follow what was there in the system and it was very difficult to think of something new [...] we would have to try harder to get this kind of abstraction.".*" – Student P37C

B. RQ2: WHAT CHALLENGES AND BENEFITS DO INSTRUCTORS PERCEIVE WHEN EMPLOYING ACTIVE LEARNING STRATEGIES WHILE TEACHING UML DIAGRAMS?

We conducted follow-up interviews with the instructors to bring more in-depth insights and answer our RQ2. The points raised during the interviews helped to understand some challenges and encouraged suggestions on how to deal with them. During data analysis, we found that the students also mentioned the challenges reported by the instructors in the focus group session. The challenges identified were: (a) students need to have prior knowledge about the content to be taught; (b) instructors need to be careful when setting the difficulty level of the scenarios used; (c) instructors need to accurately define the time to apply the strategies; and (d) instructors need to take care in developing the materials required to implement the strategies.

The first challenge (**students need to have prior knowledge about the content to be taught**) was identified during the application of three out of the five strategies: Inspection-Based strategy, Think–Pair-Square, and Similar Systems. We present the students' and instructors' comments on this challenge in Table 9.

TABLE 9. First challenge: Students need to have prior knowledge about the content to be taught.

Inspection-Based strategy: "*we had little knowledge of the notation and we were exposed to a problem that requires you to already have a more reasonable knowledge of the notation.*" – Student P08A Think-Pair-Square: "*our lack of knowledge in the notation may have completely disrupted the use of the strategy*" – Student P08A Similar Systems: *"I thought it was easy (to use the strategy), because we had already done an exercise on use cases and we already had a sense of what it was to do. (...) but if we had not done anything and were to use the strategy to model, the result would be very bad."* Student P08A
Think-Pair-Square and Inspection-Based strategies: *"I think I should have done one or two diagrams for them to see how it is before applying the strategies (Inspection Based and Think-Pair-Square strategies)."* – Instructor I01 Similar Systems: *"the students had many questions. They told me that they needed a more detailed step-by-step."* – Instructor I02

When asked about this first challenge, the instructor I01 commented that after applying the strategy s/he realized that the moment chosen for use with the students was inconvenient. The second instructor (I02) said that before applying the Similar Systems strategy, the students gave feedback that they had several questions about how to carry out the modeling of the use case. According to the instructor (I02), the students were able to better understand it after a lecture that clarified the questions about modeling.

We also observed that the **instructors need to be careful when setting the level of difficulty of the scenarios used**. This problem was reported by the students while using the Inspection-Based strategy, and the instructor observed this problem while employing the Think-Pair-Square strategy (see Table 10).

TABLE 10. Second challenge: Instructors need to be careful when setting the difficulty level of the scenarios used.

Inspection-Based strategy: *"the textual description of Inspection-Based strategy was a little confusing, it was complicated to understand, and it lacked information that we wanted.*" – Student P11A
Inspection-Based strategy: *"this strategy is completely dependent on the scenario that people are using [...] the scenario we used was a scenario that came from the industry, it was a real scenario, so it was a bit complex. The scenario had missing information, it was not as thorough as if it were an exercise, a 'toy problem', that we usually use in lectures."* – Instructor I01

Regarding the comment of the P11A, this occurred because the instructor selected a real-life development scenario as the starting point. This affected the students during modeling. When asked about the scenarios used, I01 commented that as the scenario is real, this has directly affected the use of the strategy in the classroom.

Still regarding the Inspection-Based strategy, the instructor I04 commented that as a consequence of using a real scenario, s/he had to plan another lecture to finish the application of the strategy, demonstrating the third challenge (**instructors**

TABLE 11. **Third challenge: Instructors need to properly define the time to apply the strategies.**

Positive and Negative Examples: *"I thought the exercises were very long and took two classes to finish. Maybe reducing the activity could have been easier."* – Student P05B
Inspection-Based strategy: *"the scenario was very large; it took me a long time to execute the strategy. So, I ended up spending two lectures when we were only expecting a two-hour class."* – Instructor I04   Negative Examples: "*students were not able to do the activity within the timeframe of the lesson. I had to reserve one more lecture so the students could finish the activity*" - Instructor I03

need to accurately define the time to apply the strategies– Table 11). This challenge was also perceived by I03 when employing Negative Examples, and by students using Positive and Negative Examples.

When we asked instructors about the future use of Inspection-Based strategy, I01 commented that they could adapt the scenarios according to the strategies: "*I could use simple problems, minor problems, then students would solve them faster. This happened for both strategies, especially Think-Pair-Square because it requires a much longer application time than the first one.*"

Another challenge we observed is that the **instructors need to be careful in developing the courseware needed to implement the strategies** (see Table 12).

TABLE 12. **Fourth challenge: Instructors need to be careful in developing the courseware needed to implement the strategies.**

Positive and Negative Examples: *"the table with the examples contributed little since most of the team members did not even look at it [...], in very few cases the positive examples were consulted."* – Student P10B
Inspection-Based strategy: *"I did not really like to apply this strategy because of its preparation [...] To use it in modeling lecture, it is necessary: (1) to have a scenario/description of the system to model; (2) have a checklist to help students inspect the diagrams; and (3) make the material about the diagrams available for the students. It is not trivial to create all of these, especially the second point."* – Instructor I01   Positive and Negative Examples: "*my perception is that they did not follow it. I gave the assignment, some did not even see the tables with the examples [...] So, I realized they did not care much about it and they stopped reading it.*" - Instructor I02   Positive and Negative Examples: "*"I just did not like having to prepare the drawings for the problems.*" - Instructor I03

I02 developed examples in a table format when they applied the Positive Examples and Negative Examples strategies. However, the students disagreed on whether this was the most appropriate way to present these strategies. Thus, P10B reported that they did not consider these strategies useful and did not use the tables developed by the instructor. I03 employed Negative Examples in his class, and instead of using a table, they showed the negative examples in already-modeled diagrams. Although the instructor did not like having to prepare the drawings for the problems, s/he added, "*it's not a lot of effort if we think that we usually spend an almost whole day to prepare a two-hour class, so it was not that much effort.*"

In addition, instructors reported some benefits of using active learning strategies during teaching software modeling. The benefits identified were: (a) **students were feeling more engaged/challenged when using the strategies**, and (b) **strategies enable students to develop more complete and correct models** (see Table 13).

TABLE 13. **B strategies benefits perceived by instructors when employing active learning strategies.**

Students were feeling more engaged/challenged when using the strategies   Inspection-Based strategy: *"I think they loved getting their hands on the dough with this strategy (...) I have the feeling that they felt a pleasure in performing the modeling using this strategy"* – Instructor I03   Inspection-Based strategy: "*the students felt challenged and they were really motivated to do the modeling.*" - Instructor I03
Strategies enable students to develop more complete and correct models   Think-Pair-Square: *"the students had difficulty in looking at the sequence diagram and finding out in which class the method would be placed. They got mixed up in this, and the strategy helped them understand very well how to identify this."* – Instructor I01   Inspection-Based strategy: "*the strategy helped students improve the diagram shortly after finding the mistakes in their colleagues' diagrams. So, the strategy makes the student reflect, the checklist used already makes the student reflect on the diagram*" – Instructor I01

V. DISCUSSION

The results of this study encourage discussions in different directions and hold practical implications, particularly concerning instructors who seek to use teaching strategies based on active learning. The most important implication may be to emphasize the myriad ways in which instructors can employ active learning strategies in the process of teaching software modeling.

Regarding the results of the study with students, we noticed positive results regarding the "factual dimension" of Bloom's taxonomy [50] and a positive "attitude toward learning" when they used active learning strategies. We also perceived that, among the strategies used, the Inspection-Based strategy was the best from the perspective of the students, considering the results in Figure 4 and our qualitative results. This may relate to the fact that this strategy allows students to receive feedback during its application. According to Cooper [56], feedback is essential for the students to understand where the models need to be improved and, most importantly, why the models are considered high quality or not.

This strategy (Inspection-Based) is based on two fundamental principles of active learning: doing and reflection. The importance of these elements is emphasized by John Dewey [57], who states that: "methods which are permanently successful in formal education . . . give the pupils something to do, not something to learn; and the doing is of such a nature as to demand thinking, or the intentional noting of connections.'' These elements are also present in Kolb's learning cycle: action and reflection. According to Schon [58], the importance of action and reflection is that they ''[help] students become proficient in a kind of reflection-in-action; and..., it involves a dialogue of coach and student that takes the form of reciprocal reflection-in-action.'' Schon [58] adds that when the strategies use this type of approach, the instructor does not need to teach the student to gain more knowledge; instead, it is enough to use this strategy as a form of training to improve learning.

Our results align with the results reported by Liu et al. [59], who showed improved student learning when using teaching strategies with feedback steps. Besides, we perceived that inspection activities helped the students to have a better understanding of the concepts taught in the class. The students reported that the feedback provided by the inspection stage allowed them to gain insight into modeling, and to find errors that were overlooked by the team members. The inspection activity improved students' understanding of the defects that should be avoided during the construction of UML diagrams. We have identified the use of checklists with students to identify defects in the models systematically.

These results are corroborated by the studies of Cooper [56], Hilburn [60] and Silva et al. [47]. In all these studies, we identified several positive results of the use of inspection in Software Engineering Education, for example, the reduction of the number of defects in the models elaborated by the students. The authors realized that, after performing the inspection phase, students could better understand the concepts, and that identifying errors prevents students from repeating them in future modeling. When combining strategies that are focused only on ''doing'' and ''reflection'' strategies, students tend to achieve learning objectives during the teaching of software modeling more effectively. Shekhar et al. [61] commented that when the strategy allows students to interact with each other, medium to high student engagement levels was observed. On the other hand, instances with low instruction participation consistently received low student engagement.

However, concerning Similar Systems, we observed a high rate of disagreement (and neutral feedback). This may be associated with the moment the strategy was applied: the timing. In the case of this strategy, the students of Class B had been previously exposed to two teaching strategies (Positive Examples and Negative Examples). Apart from that, students commented that using this strategy was very difficult because of the high level of abstraction it requires –students need to abstract the scenario from an already developed system first, and then create the model. Students suggested using the strategy to train the knowledge about the content, not directly as a form of teaching the content. We believe that this is because when students had not learned the content completely, they possibly did not perform correct modeling at first.

Although the literature states that using different strategies in the teaching process improves students' cognitive levels, as reported in Silva et al. [47] and Vujović et al. [62], in our results we realized that this might not be the case for all strategies or contents. Perhaps this has been influenced by the cultural context of our students and their educational culture since some students took a few courses that adopt active learning strategies, and others are used to learning passively [63]. These may have influenced the students' perceptions within the learning environment. We also observed that instructors may not be ready to use new teaching strategies. We identified some of the challenges they had in introducing these strategies in their classrooms through interviews. These challenges may hinder the application of strategies when teaching software modeling. Therefore, it is necessary to train the instructors beforehand, since their inability may negatively impact the strategy's implementation.

Constructivism implies that students must build new knowledge about their prior experience, which in the context of software modeling is important. However, the students' previous knowledge of the content, to which they began to formulate new knowledge, was not fully consolidated [64]. Therefore, what we perceive in the first challenge (**students need to have prior knowledge about the content to be taught**) is that students first needed to be more involved in the learning processes. We noticed that it is not appropriate to use these strategies early in the learning because it can confuse the students. Therefore, we recommend that, as a prerequisite for using these three strategies, the instructor needs to teach the concepts and to demonstrate practical examples so the student understands the notations of the diagram taught. Thus, active learning strategies would better support knowledge building and help the students during modeling [65]. The active learning strategies also address the issue of students having different knowledge levels because each student finds the best way to work and learn to contribute to the expected final result [19].

According to Fioravanti et al. [46], students are often unable to perform the activities because of the complexity of the scenarios used by the instructors. Therefore, concerning the challenge **instructors need to be careful when setting the level of difficulty of the scenarios**, since students may be confused with high complexity scenario, and fail to complete the modeling activity. This, in turn, overwhelms the instructors with various requests for help and assistance. This may occur even if the students are working in small groups in which collaboration and interaction among team members are present, as students do not feel as safe when the scenario is complex or based on a real system.

Highly related to this, we found that the **instructors need to accurately define the time to apply the strategies** (challenge 03). According to Port and Kazman [66], it is difficult to predict the amount of time required to carry out active learning strategies. The authors commented that ''this seems to be highly variable and highly idiosyncratic (perhaps unmanageable because it depends on the skill levels of the individuals in the lectures and these vary widely." Almost all students are concerned with stress due to time and delivery of the modeling [68], and, according to Aksit et al. [68], students may complain that they were working under time pressure, hampering the final modeling. Students thus tend to prefer theoretical lessons over classes that use these strategies [69]. As a suggestion to this challenge, if the instructors wish to apply one of the strategies in only one class (lasting 40 minutes), it is best to use a "toy scenario," whether the instructors create it or took it from a book, or, for example, to have students modeling only part of a particular scenario. Besides, in strategies whose steps are well defined, such as in Think-Pair-Square, the instructors must set a time limit for students to complete a first version of the model.

Finally, by employing these strategies in the classroom, instructors face the inevitable task of rethinking each class to make it more interactive, practical, engaging, and student-centered [66]. Therefore, the instructor needs to put more effort into implementing these strategies (which leads to more pressure). There is more work, so **the instructors need to be careful when developing the courseware required to implement the strategies** (challenge 04). This step is paramount since, according to Aksit et al. [68], most students complain about the lack of materials needed to apply the strategies. This could be a barrier to the implementation of some activities involving active learning, but certainly not for all, which results in an even higher workload for instructors. In this way, we realize that some strategies need some preparation to implement them. This may discourage some instructors from applying these strategies, because it requires more time to prepare the classes and it may not always elicit positive feedback from students, causing frustration and a temporary impediment to the instructor's adoption of new teaching strategies. Thus, we suggest it could be interesting to create an online repository for the instructor to make available the materials and information necessary for other instructors to apply these strategies in their classes.

Prior works pointed a range of issues and factors on student's resistance on active learning strategies. We noticed that the results of the current study are also in parallel to the works presented by Baker and Hill [67] and Aksit et al. [68]. Similarly, in this study, the authors emphasized the following challenges: resistance to innovation; lack of interest; unacceptable learning methods to students; preference for traditional teaching methods; reluctance to change teaching approach; age-profile (that is, soon-to-retire academics are reluctant to change). According to Kenney-Kennicutt and Simpson [70], this student's resistance is the result of the shift in thinking about who has responsibilities for what actions and processes in the class. In this context, researchers have offered strategies to instructors to acknowledge and overcome this resistance, including active listening and response to student concerns [70].

VI. LIMITATION

Although we analyzed data from various sources and different classrooms, we likely did not discover all students' perceptions about teaching strategies, and we probably did not uncover all the challenges of instructors or provide full explanations of these challenges. We are aware that each classroom that applies the strategy has its singularities and that the universe of teaching software modeling using active learning strategies is massive. This means that the perceptions reported by the students and the identified challenges may differ according to the strategy or the class. This variety in the answers of the students may have been influenced by factors that directly affect the actual learning experience, for example, level of knowledge of the instructor (novice, beginner, experienced, or expert), and both student's and instructor's amount of orientation, training, and instruction with active learning strategies. Our strategy to consider different methods and different classes with different student profiles aimed to alleviate this problem by identifying the perceptions and challenges from multiple perspectives. Besides that, providing learners with a variety of active learning strategies will address their many learning needs based on their learning styles and placement in their learning process.

Another threat to the results' validity is the subjectivity of the classifications of the collected data. To mitigate this threat, we used an approach in which all analysis was based on the data collected. Besides, we exhaustively discussed the process of analysis along with two other researchers to promote better validation of interpretations through agreement.

During the focus group, when asking students to note their perceptions on colored papers and read them to other students, some of these perceptions could have been influenced in some way. To minimize this threat, we encouraged the students to provide their opinions and comments as sincerely as possible. Besides that, the physical organization of the classroom used and a large number of simultaneous discussions between teams in the focus groups probably hampered collecting richer data. In that sense, we asked the students to make only one comment at a time. We could have split the discussions into other focus groups sessions, but we could not make sure the participants would be available or committed to participate again.

In addition, during the interview, instructors were asked to answer questions related to the application of the strategies they chose. The interview script was semi-structured and therefore contained open items to capture the perceptions of the teachers on the strategies. To minimize this threat, the interview script was evaluated by two researchers. After the evaluation, a pilot interview was conducted with an instructor with previous experience in modeling teaching, to verify if the script would reach its goal. The results of the interview were evaluated together with another researcher, and there was no need to improve the script.

VII. CONCLUSION

The use of active learning strategies has been gaining prominence in computing courses. In this paper, we carried out several case studies to understand the influence of these strategies in the teaching and learning process of modeling UML diagrams. We collected qualitative data through questionnaires and focus group sessions conducted with students and semi-structured interviews with instructors. From the analysis of students' perceptions, we could observe that some strategies were more useful for students (Inspection-Based strategy, for example), while others were considered more complicated and not very useful (Similar Systems, for example).

Qualitative analysis allowed us to identify fifteen factors (benefits and difficulties) that influence the learning that employs active learning strategies, which were organized as follows: points of view during modeling (02 benefits and 01 difficulty); collaborative learning (03 benefits); support during modeling (04 benefits and 02 difficulties); feedback provided to students (01 benefit and 01 difficulty), and abstraction and problem solving modeling (02 benefits and 01 difficulty). The benefits and difficulties that we identified can help instructors choose the strategies to apply and students to achieve a specific learning objective.

We also identified some challenges reported by instructors when using these strategies: (i) students need to have prior knowledge about the content; (ii) instructors need to take care when setting the level of difficulty of the scenarios used; (iii) instructors need to properly define the time to apply the strategies; (iv) and instructors need to carefully develop the materials needed to implement the strategies. For each challenge, we suggest strategies to overcome them.

From the results of this study, some questions arose: did the size of the teams influence the results? Did team-formation influence group modeling? Would pair modeling be more effective, since decisions do not depend on consensus among many people? Were student profiles and learning styles factors that affected the study? As future work, we intend to carry out a complete evaluation to measure the effectiveness of the strategies concerning the learning process. In this paper, we aimed to analyze the various aspects of strategies, perceptions about motivation and perceived learning, and so on. We intend to carry out new studies with more students and in other contexts (different courses, different states, private and public educational institutions, night and/or day courses) to identify new factors that may influence the use of these strategies; in addition to new challenges faced by instructors when using these strategies, as well as suggest alternatives for how to minimize these challenges.

ACKNOWLEDGMENT

The authors would like to thank all the students who participated in this research. We thank the financial support from Northern Arizona University.

REFERENCES

[1] O. Hazzan and J. Kramer, "Abstraction in computer science & software engineering: A pedagogical perspective," *Frontier J.*, vol. 4, no. 1, pp. 6–14, 2007.

[2] V. Y. Sien, "An investigation of difficulties experienced by students developing unified modelling language (UML) class and sequence diagrams," *Comput. Sci. Educ.*, vol. 21, no. 4, pp. 317–342, 2011. doi: 10.1080/08993408.2011.630127.

[3] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*, 2nd ed. London, U.K.: Pearson, 2004.

[4] H. Störrle, "How are conceptual models used in industrial software development?: A descriptive survey," in *Proc. 21th Int. Conf. Eval. Assessment Softw. Eng. (EASE)*, Karlskrona, Sweden, Jun. 2017, pp. 160–169.

[5] Z. Shen, S. Tan, and K. Siau, "Challenges in learning unified modeling language: From the perspective of diagrammatic representation and reasoning," *Commun. Assoc. Inf. Syst.*, vol. 43, no. 1, pp. 545–565, Sep. 2018. doi: 10.17705/1CAIS.04330.

[6] K. Siau and P.-P. Loo, "Identifying difficulties in learning UML," *Inf. Syst. Manage.*, vol. 23, no. 3, pp. 43–51, Dec., 2006. doi: 10.1201/1078.10580530/46108.23.3.20060601/93706.5.

[7] N. Bolloju and F. S. K. Leung, "Assisting novice analysts in developing quality conceptual models with UML," *Commun. ACM*, vol. 49, no. 7, pp. 108–112, Jul. 2006. doi: 10.1145/1139922.1139926.

[8] Z. Ma, "An approach to improve the quality of object-oriented models from novice modelers through project practice," *Frontiers Comput. Sci.*, vol. 11, no. 3, pp. 485–498, Jun. 2017. doi: 10.1007/s11704-016-5164-8.

[9] R. Szmurło and M. Śmiałek, "Teaching software modeling in a simulated project environment," in *Proc. 9th Int. Conf. Model Driven Eng. Lang. Syst.*, Berlin, Germany, 2006, pp. 301–310.

[10] K. Al-Tahat, "An innovative instructional method for teaching object-oriented modelling," *Int. Arab J. Inf. Technol.*, vol. 11, no. 6, pp. 540–549, Nov. 2014.

[11] J. Chen, H. Lu, L. An, and Y. Zhou, "Exploring teaching methods in software engineering education," in *Proc. 4th Int. Conf. Comput. Sci. Educ.*, Nanning, China, Jul. 2009, pp. 1733–1738.

[12] N. Capuano, A. Pierri, S. Salerno, and G. R. Mangione, "Combining individualization and intuitive guided learning through compound learning resources," in *Proc. 6th Int. Conf. Complex, Intell., Softw. Intensive Syst.*, Palermo, Italy, Jul. 2012, pp. 905–910.

[13] R. Holmes, M. Allen, and M. Craig, "Dimensions of experientialism for software engineering education," in *Proc. 40th ICSE-SEET*, Gothenburg, Sweden, 2018, pp. 31–39.

[14] S. A. A. de Freitas, W. C. M. P. Silva, and G. Marsicano, "Using an active learning environment to increase students' engagement," in *Proc. IEEE 29th Int. Conf. Softw. Eng. Educ. Training*, Dallas, TX, USA, Apr. 2016, pp. 232–236.

[15] T. Briggs, "Techniques for active learning in CS courses," *J. Comput. Sci. Colleges*, vol. 21, no. 2, pp. 156–165, Dec. 2005.

[16] M. Prince, "Does active learning work? A review of the research," *J. Eng. Educ.*, vol. 93, pp. 223–231, Feb. 2013. doi: 10.1002/j.2168-9830.2004.tb00809.x.

[17] C. C. Bonwell and J. A. Eison, "Active learning: Creating excitement in the classroom. 1991 ASHEERIC higher education reports," George Washington Univ., Washington, DC, USA, ED336049, 1991. [Online]. Available: https://eric.ed.gov/?id=ED336049

[18] C. E. Hmelo-Silver, "Problem-based learning: What and how do students learn?" *Educ. Psychol. Rev.*, vol. 16, no. 3, pp. 235–266, Sep. 2004. doi: 10.1023/B:EDPR.0000034022.16470.f3.

[19] S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth, "Active learning increases student performance in science, engineering, and mathematics," *Proc. Nat. Acad. Sci. USA*, vol. 111, no. 23, pp. 8410–8415, Jun. 2014. doi: 10.1073/pnas.1319030111.

[20] J. Michael, "Where's the evidence that active learning works?" *Adv. Physiol. Educ.*, vol. 30, no. 4, pp. 159–167, Dec. 2006. doi: 10.1152/advan.00053.2006.

[21] W. W. Schilling, Jr., and M. J. Sebern, "Teaching software engineering: An active learning approach," in *Proc. 119th Amer. Soc. Eng. Educ.*, San Antonio, TX, USA, 2012, pp. 436–442.

[22] J. Knobloch, J. Kaltenbach, and B. Bruegge, "Increasing student engagement in higher education using a context-aware Q&A teaching framework," in *Proc. 40th ICSE-SEET*, Gothenburg, Sweden, 2018, pp. 136–145.

[23] R. R. Nunes, D. Pedrosa, B. Fonseca, H. Paredes, J. Cravino, L. Morgado, and P. Martins, "Enhancing students' motivation to learn software engineering programming techniques: A collaborative and social interaction approach," in *Proc. 9th Int. Conf. Universal Access Hum.-Comput. Interact.*, Los Angeles, CA, USA, 2015, pp. 189–201.
[24] I. Hadar and E. Hadar, "An iterative methodology for teaching object oriented concepts," *Inform. Educ.*, vol. 6, no. 1, pp. 67–80, Jan. 2007.
[25] J. Kontio, L. Lehtola, and J. Bragge, "Using the focus group method in software engineering: Obtaining practitioner and user experiences," in *Proc. ISESE*, Redondo Beach, CA, USA, Aug. 2004, pp. 271–280.
[26] A. L. Strauss and J. Corbin, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 2nd ed. London, U.K.: SAGE, 1998.
[27] L. T. W. Agner and T. C. Lethbridge, "A survey of tool use in modeling education," in *Proc 20th MODELS*, Austin, TX, USA, Sep. 2017, pp. 303–311.
[28] G. Liebel, O. Badreddin, and R. Heldal, "Model driven software engineering in education: A multi-case study on perception of tools and UML," in *Proc. 30th CSEE&T*, Savannah, GA, USA, Nov. 2017, pp. 124–133.
[29] O. Badreddin, R. Khandoker, A. Forward, O. Masmali, and T. C. Lethbridge, "A decade of software design and modeling: A survey to uncover trends of the practice," in *Proc. 21th MODELS*, Copenhagen, Denmark, 2018, pp. 245–255.
[30] C. G. von Wangenheim, R. Savi, and A. F. Borgatto, "SCRUMIA—An educational game for teaching SCRUM in computing courses," *J. Syst. Softw.*, vol. 86, no. 10, pp. 2675–2687, Oct. 2013. doi: 10.1016/j.jss.2013.05.030.
[31] R. Adams, D. Evangelou, L. English, A. D. de Figueiredo, N. Mousoulides, A. L. Pawley, C. Schiefellite, R. Stevens, M. Svinicki, J. M. Trenor, and D. M. Wilson, "Multiple perspectives on engaging future engineers," *J. Eng. Educ.*, vol. 100, no. 1, pp. 48–88, Jan. 2011. doi: 10.1002/j.2168-9830.2011.tb00004.x.
[32] H. A. Aglan and S. F. Ali, "Hands-on experiences: An integral part of engineering curriculum reform," *J. Eng. Educ.*, vol. 85, no. 4, pp. 327–330, Oct. 1996. doi: 10.1002/j.2168-9830.1996.tb00252.x.
[33] A. Talley, J. Lindsey, D. Jensen, and K. Wood, "Development and assessment of active learning products to enhance engineering mechanics courses," in *Proc. 115th Amer. Soc. Eng. Educ.*, 2007, pp. 1–8.
[34] R. W. Welch and J. L. Klosky, "An online database and user community for physical models in the engineering classroom," *Adv. Eng. Educ.*, vol. 1, no. 1, pp. 1–25, Jan. 2007.
[35] S. Dennis, M. Bowe, J. Ball, and D. Jensen, "A student-developed teaching demo of an automatic transmission," in *Proc. Amer. Soc. Eng. Educ.*, 2001, pp. 104.1–104.13.
[36] J. S. Linsey, I. Tseng, K. Fu, J. Cagan, K. L. Wood, and C. Schunn, "A study of design fixation, its mitigation and perception in engineering design faculty," *ASME J. Mech.*, vol. 132, no. 4, Apr. 2010, Art. no. 041003. doi: 10.1115/1.4001110.
[37] J. Kerrigan, "Active learning strategies for the mathematics classroom," *College Teach.*, vol. 66, no. 1, pp. 35–36, Jan. 2018. doi: 10.1080/87567555.2017.1399335.
[38] C.-L. Lai and C.-J. Hwang, "A self-regulated flipped classroom approach to improving students' learning performance in a mathematics course," *Comput. Educ.*, vol. 100, pp. 126–140, Sep. 2016. doi: 10.1016/j.compedu.2016.05.006.
[39] O. Karamustafaoglu, "Active learning strategies in physics teaching," *Energy Educ. Sci. Technol.*, vol. 1, no. 1, pp. 27–50, Jan. 2009.
[40] M. Baser, "Promoting conceptual change through active learning using open source software for physics simulations," *Australas. J. Educ. Technol.*, vol. 22, no. 3, pp. 336–354, Jan. 2006. doi: 10.14742/ajet.1290.
[41] L. Junhua, "Innovations in software engineering education: An experimental study of integrating active learning and design-based learning," in *Proc. Amer. Soc. Eng. Educ.*, Denver, CO, USA, 2013, p. 1.
[42] R. D. Knight, *Five Easy Lessons: Strategies for Successful Physics Teaching*. San Francisco, CA, USA: Addison-Wesley, 2004.
[43] E. Exposito, "yPBL methodology: A problem-based learning method applied to software engineering," in *Proc. EDUCON*, Madrid, Spain, Apr. 2010, pp. 1817–1823.
[44] A. García-Holgado, F. J. García-Peñalvo, and M. J. Rodríguez-Conde, "Pilot experience applying an active learning methodology in a software engineering classroom," in *Proc. EDUCON*, Tenerife, Spain, Apr. 2018, pp. 940–947.
[45] G. Scanniello and U. Erra, "Distributed modeling of use case diagrams with a method based on think-pair-square: Results from two controlled experiments," *J. Vis. Lang. Comput.*, vol. 25, no. 4, pp. 494–517, Aug. 2014. doi: 10.1016/j.jvlc.2014.03.002.
[46] M. L. Fioravanti, B. Sena, L. N. Paschoal, L. R. Silva, A. P. Allian, E. Y. Nakagawa, S. R. S. Souza, S. Isotani, and E. F. Barbosa, "Integrating project based learning and project management for software engineering teaching: An experience report," in *Proc. 49th SIGCSE*, Baltimore, MD, USA, 2018, pp. 806–811.
[47] W. A. F. Silva, I. F. Steinmacher, and T. U. Conte, "Is it better to learn from problems or erroneous examples?" in *Proc. 30th CSEE&T*, Savannah, GA, USA, Nov. 2017, pp. 222–231.
[48] W. Silva, I. F. Steinmacher, and T. U. Conte, "Technical report on systematic literature mapping on the difficulties related to the learning of UML diagrams," Federal Univ. Amazonas, Manaus, Brazil, USES Res. Group, Tech. Rep. TR-USES-2017-019, Dec. 2019. [Online]. Available: http://uses.icomp.ufam.edu.br/relatorios-tecnicos/
[49] G. Sindre and D. L. Moody, "Evaluating the effectiveness of learning interventions: An information systems case study," in *Proc. ECIS*, 2003, pp. 80–97.
[50] L. W. Anderson, D. R. Krathwohl, P. W. Airasian, K. A. Cruikshank, R. E. Mayer, P. R. Pintrich, J. Raths, and M. C. Wittrock, *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. White Plains, NY, USA: Longman, 2001.
[51] V. Ekahitanond, "Promoting university students' critical thinking skills through peer feedback activity in an online discussion forum," *Alberta J. Educ. Res.*, vol. 59, no. 2, pp. 247–265, Apr. 2013.
[52] W. Orawiwatnakul and S. Wichadee, "Achieving better learning performance through the discussion activity in Facebook," *Turkish Online J. Educ. Technol.*, vol. 15, no. 3, pp. 1–8, Jul. 2016.
[53] E. Colucci, "'Focus groups can be fun': The use of activity-oriented questions in focus group discussions," *Qualitative Health Res.*, vol. 17, no. 10, pp. 1422–1433, Dec. 2008. doi: 10.1177/1049732307308129.
[54] C. B. Seaman, "Qualitative methods in empirical studies of software engineering," *Trans. Softw. Eng.*, vol. 25, no. 4, pp. 557–572, Jul./Aug. 1999. doi: 10.1109/32.799955.
[55] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Softw. Eng.*, vol. 14, no. 2, pp. 131–164, Apr. 2009. doi: 10.1007/s10664-008-9102-8.
[56] K. Cooper, S. Liddle, and S. Dascalu, "Experiences using defect checklists in software engineering education," in *Proc. CAINE*, 2005, pp. 402–409.
[57] J. Dewey, *Democracy and Education: An Introduction to the Philosophy of Education*. New York, NY, USA: Macmillan, 1916.
[58] D. Schon, *Educating the Reflective Practitioner*, San Francisco, CA, USA: Jossey-Bass, 1987.
[59] E. Z.-F. Liu, S. S. J. Lin, C.-H. Chiu, and S.-M. Yuan, "Web-based peer review: The learner as both adapter and reviewer," *IEEE Trans. Educ.*, vol. 44, no. 3, pp. 246–251, Aug. 2001. doi: 10.1109/13.940995.
[60] T. B. Hilburn, M. Towhidnejad, and S. Salamah, "Read before you write," in *Proc. 24th CSEE&T*, Honolulu, HI, USA, 2011, pp. 371–380.
[61] P. Shekhar, M. Prince, C. Finelli, M. Demonbrun, and C. Waters, "Integrating quantitative and qualitative research methods to examine student resistance to active learning," *Eur. J. Eng. Educ.*, vol. 44, nos. 1–2, pp. 6–18, Jan. 2019. doi: 10.1080/03043797.2018.1438988.
[62] V. Vujović, M. Maksimović, and B. Perišić, "The different active learning strategies in Software Engineering and their effectiveness," in *Proc. 7th ICERI*, Seville, Spain, 2014, pp. 3183–3193.
[63] L. B. Nilson, *Teaching at Its Best: A Research-Based Resource for College Instructors*, 4rd ed. San Francisco, CA, USA: Wiley, 2016.
[64] P.-M. Nguyen, C. Terlouw, and A. Pilot, "Culturally appropriate pedagogy: The case of group learning in a confucian heritage culture context," *Intercultural Educ.*, vol. 17, no. 1, pp. 1–19, Aug. 2006. doi: 10.1080/14675980500502172.
[65] E. von Glasersfeld, *Radical Constructivism: A Way of Knowing and Learning*. Abingdon, U.K.: Routledge, 2013.
[66] D. Port and R. Kazman, "Laptop enabled active learning in the software engineering classroom: An experience report," in *Proc. 20th CSEET*, Dublin, Ireland, Jul. 2007, pp. 262–274.
[67] E. Baker and S. Hill, "Investigating student resistance and student perceptions of course quality and instructor performance in a flipped information systems classroom," *Inf. Syst. Educ. J.*, vol. 15, no. 6, pp. 17–26, Nov. 2017.
[68] F. Aksit, H. Niemi, and A. Nevgi, "Why is active learning so difficult to implement: The Turkish case," *Austral. J. Teacher Educ.*, vol. 41, no. 4, pp. 94–109, Apr. 2016. doi: 10.14221/ajte.2016v41n4.6.

[69] S. Tharayil, M. Borrego, M. Prince, K. A. Nguyen, P. Shekhar, C. J. Finelli, and C. Waters, "Strategies to mitigate student resistance to active learning," *Int. J. STEM Educ.*, vol. 5, Jan. 2018, Art. no. 7. doi: 10.1186/s40594-018-0102-y.

[70] W. Keeney-Kennicutt, A. B. Gunersel, and N. Simpson, "Overcoming student resistance to a teaching innovation," *Int. J. Scholarship Teaching Learn.*, vol. 2, no. 1, p. 5, 2008. doi: 10.20429/ijsotl.2008.020105.

WILLIAMSON SILVA is currently pursuing the Ph.D. degree in informatics with the Institute of Computing of the Federal University of Amazonas (UFAM). His research interests include software engineering education, teaching software modeling, active learning strategies, and related topics. He is also a member of the USES (usability and software engineering) Research Group.

IGOR STEINMACHER received the Ph.D. degree in computer science from the University of São Paulo. He is currently as an Assistant Professor with the School of Informatics, Computing, and Cyber Systems, Northern Arizona University. His research interest includes human aspects of software engineering and related topics.

TAYANA CONTE is currently a Professor of software engineering with the Computing Institute of the Federal University of Amazonas (UFAM), Brazil. She lead the USES (usability and software engineering) Research Group.