Leveraging Teaching Methods to Overcome Challenges in DevOps Education

V. Shobha Rani[1], Dr. A. Ramesh Babu [2]

[1]Research Scholar, Department of Computer Science, Chaitanya (Deemed to be University), Warangal.
Email: shobhareddy19@gmail.com

[2] Professor, Department of Computer Science, Chaitanya (Deemed to be University), Warangal.

Email: rameshadloori@gmail.com

ABSTRACT

DevOps comprises a practice that connects the software development and operations, allowing for faster, more dependable software product delivery. DevOps has become a highly demanded skill in the software industry, and many universities have started to offer DevOps courses as part of their software engineering curricula. However, teaching DevOps poses several challenges, such as selecting appropriate tools and technologies, integrating theory and practice, balancing breadth and depth, assessing learning outcomes, and providing feedback to students. In this research, we aim to address the question of how to design and evaluate effective DevOps courses for software engineering students. Our research method is a combination of different techniques. We review the literature systematically, and we interview and survey people who teach or practice DevOps Based on interviews, we analyzed how teaching methods relate to DevOps educational challenges. Our findings revealed that the most successful approaches are the project-based and collaborative learning.

Key words—DevOps, teaching methods, challenges, DevOps Education, mixed method

I. INTRODUCTION

DevOps (Development and Operations) emerges as a natural progression of the agile movement [1]. Its primary objective is to reduce conflicts, approximate duties, and increase communication between Development and Operations teams [2]. This Agile mindset is critical for firms to remain competitive in the technological business, fulfilling the need for rapid, durable, and secure outputs involving distributed systems at scale. Recent research indicates that the evolution of DevOps processes in industry organizations correlates with company performance. Since 2009, several significant firms have adopted DevOps, including Yahoo, Flickr, Facebook, Etsy, and Netflix[3]. Companies like Amazon and Netflix deploy hundreds of times each day using DevOps practices. As a result, mastering DevOps processes has become a required skill in many IT domains[4].

DevOps is tough to understand in the industry or on large, complex projects [5]. To study these elements further, DevOps research with an educational focus is required. Investing in DevOps training is an effective approach to reduce industry demand. From an educational standpoint, the educator's responsibility is to assist students through the learning process by employing effective methods for teaching and providing an appropriate atmosphere for learning [6]. Although there is no dominating strategy, depending on the educational setting, certain approaches are more effective than others [7]. For example, the project-based learning teaching method is commonly regarded as an essential component of Software Engineering. (SE) Education [8-10].

Educators and practitioners agree that the standard lecture format is insufficient for teaching SE [11]. The main disadvantages of this strategy include student inactivity, tiring lectures, one-way communication, and rapid forgetfulness [12], [13]. In this regard, tackling teaching from many perspectives stems from the notion that everyone learns differently. As a result, teaching must be viewed as a dynamic variable. The attitude to diversity in pupils has an impact on the variety of teaching [11]. Another option for educators is to integrate many teaching methods and approaches in a single course. This is not a novel concept, as evidenced by the combination of academic (lectures) and practical (lab) sessions. The issue, however, is to defend all teaching approaches to into one course [14].

Ferino et al. [5] presented an early study of DevOps Education training approaches. They examined 18 papers from a systematic literature analysis of DevOps teaching experience published up to 2019. This study kicked off research into DevOps teaching approaches. However, because this is a preliminary study, more research is needed to validate and improve educators' knowledge of DevOps teaching approaches.

In this paper, we explore how DevOps courses teach their students and how they deal with the problems found in the DevOps Education sector. We plan to answer these research questions:

RQ1: What methodologies and instructional methods are used in DevOps courses?

RQ2: How could DevOps course challengesis addressed by instructional methods?

II. RELATED WORK

Ferino et al. [5] give a first analysis of teaching methods discovered in the literature, demonstrating how to assist and reduce obstacles encountered in DevOps training. It pertains to this work because it also seeks for DevOps training approaches, providing 13 of them. They dug deeper into the educational support tool teaching technique, concentrating on how it relates to educational DevOps difficulties. In contrast, our study includes a total of 18 teaching strategies found through interviews with DevOps educators. Furthermore, we confirmed the results by using internal validation through member verification and external comparison with the Ferino et al. [5] study.

Grotta et al. [11] reviewed the literature on the performance of students in DevOps who have received Information System (IS) training. They discovered the correlation between teaching DevOps and both professional and academic advancement. It highlights the intimate relationship between DevOps and project-based learning. That work is connected to ours because it investigates teaching approaches for DevOps. However, their study specifically focuses on connecting DevOps to the Information Systems domain, whereasour study is focused on revealing teaching methods and how they pertain to the difficulties faced in DevOps courses.

Fernandes et al. [15] examine interviews with DevOps instructors from around the world, gathering difficulties and solutions based on their experiences from existing DevOps courses. This work is linked to ours because it looks into issues in DevOps education. Our approach is distinct in that it focuses on identifying instructional strategies and connecting them to these issues.

III. METHODOLOGY

The work conducted in this study can be divided into two phases.

(i) Identifying the teachingmethodologies for DevOps; and
(ii) Creating association links between methods of teaching to challenges.

An interview study done with 20 DevOps educators, serves as the foundation for our analyses. These educators have extensive expertise teaching computer courses. They discussed their experiences teaching DevOps courses throughout the interviews. All of these courses have a DevOps focus and address various DevOps components [16] such as Runtime and Delivery. Based on their interview transcripts, we identify the teaching approaches and connect them to the issues identified by Fernandes et al.We interviewed 15 male educators and 5 female educators, who are referred to as P1-P20 throughout this research, where P15 refers to the fifteenth interview, for example.

Choosing Teaching Methods

Based on the 20 interview transcripts, the initial part of this endeavor is to identify teaching methods. We are using Westwood et al.'s [17] definition of a teaching method: a set of principles, processes, or techniques used by teachers to facilitate desired learning in students [18], [19].

The subsections that follow go over each stage in further detail.

1) Extraction of Data:

The analysis begins by extracting teaching methods from interviews' transcripts. For instance, the interviewee organizes students into teams, teaching collaborative learning and team building. Additionally, Interviewee P14, she uses problem-based learning (PBL) and flipped classroom in her DevOps course, demonstrating the effectiveness of these teaching methods in enhancing student learning.

2) Data Analysis:

The study used open coding to identify and categorize teaching methods in extracted data. It was critical to employ codification because teaching approaches were not always precisely stated. The study also examined interview transcripts to clear up the adoption of teaching methodology, such as problem-based learning. If a text snippet cannot confirm the method's use, it is not considered to be implemented.

3) Participant feedback (member-check):

Survey was conducted with DevOps educators to evaluate their teaching methods. The survey included closed-ended and open questions, with open questions allowing educators to share additional methods or provide links to course materials. Feedback from 6 participants (28.5%) indicating 81.25% strong with the specified teaching methods. The participants agreed on case studies,collaborative learning, educational support tools, labs,seminars, project-based learning, and collaborative learning. They also expressed neutrality towards lectures and review sessions. The survey aimed to understand the preferences of DevOps educators.

B. Linking Teaching Methods to Challenges

The study analyzed 443 links between recommendations and challenges in DevOps courses, focusing on the utilization of Problem-based Learning (PBL) asteaching approach. The researchers identified 76 links involving the same respondent, while 76 links involved different respondentsThe research focused on associations that centered around the same interviewee in both the challenge and recommendation aspects, as they demonstrated a higher degree of consistency and were primarily based on the interviewee's own experiences in course.The researchers also reviewed the found links and discussed possible conflicting opinions.

IV. RESULTS

In this section, we provide responses to the research queries posed by our research. We present the collection of artifacts [24] containing all of the data gathered and evaluated in this paper.

RQ1: What methodologies and instructional methods are used in DevOps courses?

In Figure 1, we illustrate the distribution of the 18 teaching techniques documented in the interview transcript. The top four methods, which received the most mentions, are project-based learning (cited by 18 interviewees), labs (cited by 15 interviewees), collaborative learning (cited by 13 interviewees), and lecture (cited by 12 interviewees). Conversely, the four methods with the fewest mentions are comprehensive remote learning, feedback sessions, flipped classroom, and tutorial, with only one, two, and three citations from interviewees, respectively. Our interviews revealed an average of four teaching strategies per respondent, with the highest number being seven and the lowest being two.

This paragraph discusses the identified instructional approaches. We show how educators implemented certain methods of instruction in their classes.

Table 1: Definitions of teaching approaches

PROJECT-BASED LEARNING: It centered on a project in which students complete a certain assignment [20].
COLLABORATIVE LEARNING: The students work together to exchange information and complete assignments. The teacher is an active participant, not merely storage of information, as in traditional schooling [21].
FLIPPED CLASSROOM: Traditional classroom activities become home activities (and vice versa) [22].
SEMINAR: Workshop sessions involve students working in teams to discuss predetermined issues and topics with the guidance of their instructors. Following these seminars, students are either required to deliver presentations or

compose essays on the subjects they have chosen [14], [23].
LECTURE:in educational institutions,It is the classic style of instruction, in which the instructor instructs the students directly [14], [24].
LABS: It entails completing practical exercises that explore a computer science topic, which are normally carried out in specialized rooms with computers for each student. [25].
REVIEW SESSION: In the review session, students are presented with the answer (definition), and their task is to formulate the corresponding question, essentially reversing the learning process [26].
PROBLEM-BASED LEARNING: This is a method of structuring and delivering courses where problems serve as the key motivators and central points for students' engagement and learning activities [27].
PERSONALIZED LEARNING: It describes several instructional methodologies targeted at satisfying individual learning demands [28].
EXPERIENTIAL LEARNING: It involves the acquisition of knowledge from practical experiences and the subsequent application of that knowledge in real-world situations [29].
EDUCATIONAL SUPPORT TOOL:A teaching or learning aid or integrated environment [30].
COMPREHENSIVE DISTANCE LEARNING: This is a purposeful instructional method that is selected proactively, allowing time for careful planning and preparation to ensure the quality and inclusiveness of the learning experiences for all students [31].
AGILE PROCESS: Use of Agile activities such as sprints and scrum planning throughout course execution [32].
CASE STUDY: Practical examples encourage students to apply classroom knowledge in real life [33].
FEEDBACK SESSION: The educator provides extensive feedback for the submitted assignments, which focus on the areas where the majority of people performed well and the flaws that were detected. [34].
MENTORING: This method involves enhancing teaching through the involvement of a mentor in educational activities. The mentor assists in organizing the team's work and ensuring that the team achieves its expected educational outcomes [35].
EXAMPLE-BASED LEARNING: Its foundation is built on presenting worked examples that provide a written account of how an issue should or can be solved [36].
TUTORIAL:A video or documentation that introduces the general to theoretical leader or technical topics as well as configuration steps [37].

Project-based Learning:

The teaching method P15 involves students working on a concrete project, incrementally incorporating DevOps practices. The educator may provide a starter project or an open-source project for students to improve by adding DevOps practices.

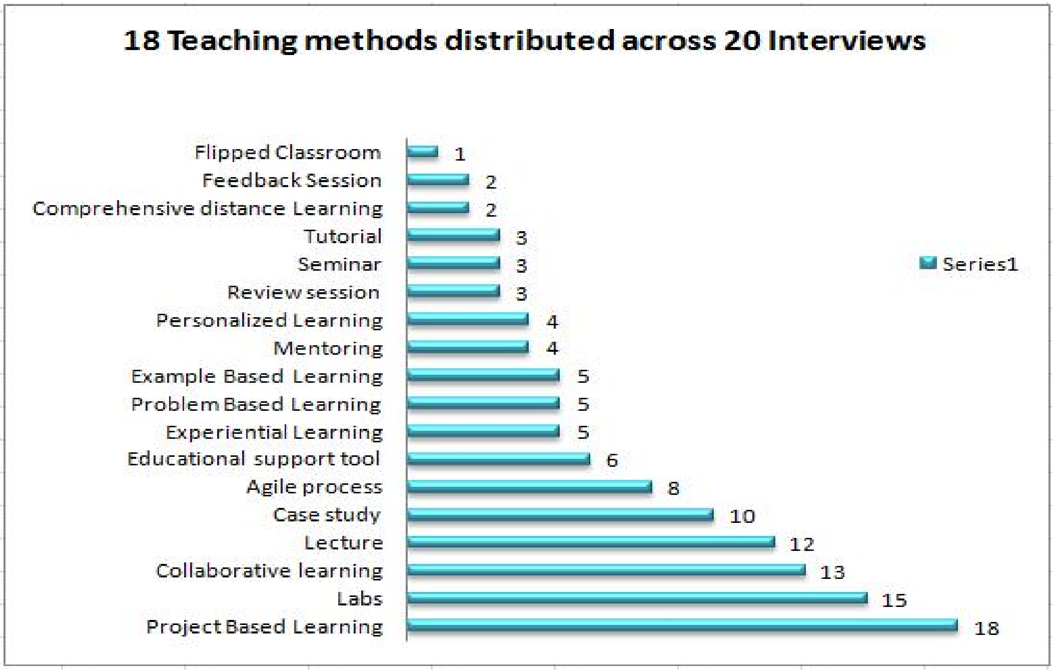
The text proposes an incremental software project development approach to teach students how to deal with complexity. It suggests using various DevOps-related modules, encouraging innovative projects, and focusing on other computer science areas like artificial intelligence. The course also encourages students to contribute to open-source projects and industrial projects to understand realistic scenarios. The project requirements are planned, considering programming languages and associated tools. The course also provides examples of projects to help students understand how to meet requirements. The text also highlights the significance of using a project-based approach instead of traditional exams, using the Net Promoter Score (NPS) for feedback.

Labs:P3 and P13 demonstrate DevOps concepts using various tools, including free and open-source technologies. P15 emphasizes practicing concepts with multiple tools, while P13 adopts open-source tools for realistic practice. P18 automates lab creation using snapshots, facilitating large classes with personalized attendance. P10 and P14 face challenges with updating tools, but teaching assistants with proficiency in tools reduce preparation effort.

Collaborative learning: Collaboration is crucial in DevOps, encouraging the collaboration and team work among development and operation teams. P8 emphasizesnecessity of industrial collaboration. P2 tracks student collaboration using a Continuous Integration tool, while P6 uses group problems and projects. P12, P17, and P8 teach collaborative work through team projects, highlighting problem sharing and considering differences in team environments. P8 uses pair programming and a collaborative slack workspace, emphasizing team members' responsibility.

Other teaching methods: P20 uses case studies from Google, Netflix, and Metasharing real-world industry experiences within the classroom. This approach encourages students to utilize Katacoda for their lessons andpromotes personalized learning.

Combining teaching methods: Teaching methods across 13 courses, categorized into three levels: **used in class, recommended, and not identified**. most of the 18 identified methodsbelongs to implemented method category, while comprehensive distance learning showing only in the suggested categories. Analysis the methodsof teaching includes 13 combinations of implemented teaching methods, 13 combinations without implementation, and 12 combinations with only implemented methods during class. Evidence of collaborative learning, project-based learning,mentoring, personalized learning, and review sessions were found in some interviews.



RQ2. How might instructional methods handle DevOps course challenges?

The study identified 44 links between challenges and recommendations related to teaching methods, including example-based learning,project-based learning, personalized learning,educational support tools, and collaboration learning. DevOps-specific challenges were identified, with most linked to unique teaching methods. The challenge of instructing DevOps principles to students lacking industrial experience was addressed through personalized learning and example-based instruction.Flipped classrooms, seminars, and tutorials were not found to have connections among challenges and recommendations.

Collaborative learning: The challenges associated with teaching Agile techniques and managing large groups of students are discussed. P8 and P10 use team-based student organization to address these challenges and promote collaboration. They also suggest personalized learning and example-based learning to help students understand conflicts between development teams and operations. P16 addresses the challenge of teaching students how to operate systems without breaking existing functionalities and understanding Continuous Delivery concepts. They recommend social coding, a collaborative learning approach that encourages students to learn from other student groups, produce ideas, exchange information, simplify problems, and resolve tasks, ultimately improving their understanding of DevOps practices.

Personalized learning: DevOps is a multidisciplinary field that can be challenging to manage due to its diverse software engineering disciplines. There exists a challenge of instructing DevOps principles to students lacking industrial experience. To solve this, educators should establish the best DevOps scope for each class and personalize learning objectives to the students' learning context. They should also reflect and improve the students' initial knowledge at the start of the courseEducators confront additional challenges when working with students from diverse backgrounds, making group collaboration harder. They suggest identifying students' needs and limitations in advance, avoiding student-specific problems, and assessing students based on their learning experiences and errors. P4 and P8 aim to foster a communication culture between students and educators. They address challenges such as remote classes, progress monitoring, and contact maintenance.

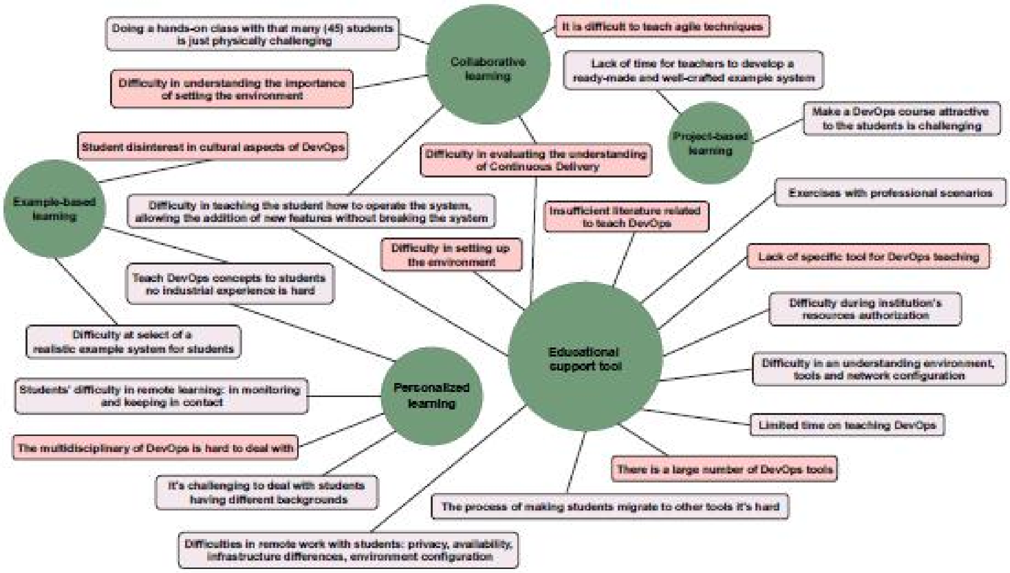

Doing a hands-on class with that many (45) students is just physically challenging
Collaborative learning
It is difficult to teach agile techniques
Lack of time for teachers to develop a ready-made and well-crafted example system
Difficulty in understanding the importance of setting the environment
Student disinterest in cultural aspects of DevOps
Project-based learning
Make a DevOps course attractive to the students is challenging
Difficulty in evaluating the understanding of Continuous Delivery
Example-based learning
Difficulty in teaching the student how to operate the system, allowing the addition of new features without breaking the system
Exercises with professional scenarios
Insufficient literature related to teach DevOps
Difficulty in setting up the environment
Lack of specific tool for DevOps teaching
Teach DevOps concepts to students no industrial experience is hard
Difficulty during institution's resources authorization
Difficulty at select of a realistic example system for students
Educational support tool
Difficulty in an understanding environment, tools and network configuration
Students' difficulty in remote learning: in monitoring and keeping in contact
Personalized learning
Limited time on teaching DevOps
The multidisciplinary of DevOps is hard to deal with
There is a large number of DevOps tools
It's challenging to deal with students having different backgrounds
The process of making students migrate to other tools it's hard
Difficulties in remote work with students: privacy, availability, infrastructure differences, environment configuration

Figure 1:Association links between DevOps Challenges and Teaching Methods.

Problem-based learning:P8 discusses the challenge of structuring the learning journey for DevOps subjects like CI and Automated Tests. They suggest using problem-based learning to teach CI, focusing on verifying changes' functionality within the codebase, rather than solely on CI tools and their capacity for innovation.

Example-based learning: The research sheds light on the hurdles of teaching DevOps concepts to students lacking industrial exposure. Frequently, students tend to fixate on tools rather than grasping the core DevOps culture, resulting in a diminished interest in its cultural dimensions. To tackle this issue, the study utilizes specific, real-world examples to illustrate that there's no one-size-fits-all solution, and each problem may require a comprehensive approach. Nevertheless, the selection of practical system examples for students can pose a challenge. The research also delves into the importance of crafting an appealing DevOps course from the students' perspective, ensuring that the projects are both challenging and of sufficient scope.

Project-based learning: The authors discuss the challenges of making a DevOps course attractive to students, including the need for interactive lectures, a lack of student creativity, and the challenge of balancing hands-on classes with labs. They suggest using teaching assistants (TA) to manage technical questions and using an online shared document for improved understanding. The authors also highlight the need for practical classes, focusing at least 80% of the classes on practical aspects, to improve students' understanding. They also highlight the gap between industry needs and university lectures, advising that industrial speakers be invited to share their experiences and that students be exposed to real-world difficulties and scenarios, strengthening their grasp of DevOps practice.

Educational support tool: Many students struggle with understanding operations, such as network configuration, due to their limited knowledge in Linux commands and operations practices. To address this, they provide an initial configuration environment incorporating needed course software, such as a containerization tool, CI/CD server, and artifact repository, via a virtual machine snapshot. Preparing courses for DevOps is challenging due to insufficient literature and insufficient material for educators. To mitigate these challenges, an integrated environment is used, allowing students to put their solutions to exercises. To make DevOps concepts more practical, we use industry-standard tools in our curriculum. Students learn by hands-on experience with tools used by companies like Google and Microsoft, allowing them to apply these concepts to real-world situations. However, some students may be hesitant to use these tools, so educators need to provide guidance and support throughout the different phases of the DevOps pipeline taught in their classes.Another problem is resource authorization, as it is difficult to reconstruct a business environment into a university one with more tighter constraints. To address this, the services of cloud providers are employed as infrastructure during the course, with students utilizing their free plans.

V CONCLUSION

A study on DevOps courses identified 18 teaching methods, with project-based learning being the most common. The study also emphasized the significance of problem-based learning in enhancing problem-solving skills. Furthermore, the study brought to light 44 connections between teaching methods and challenges in DevOps education. These included the need for educational support tools, collaborative learning, personalized learning, example-based learning, and project-based learning. The research also delved into potential avenues for further study, focusing on new teaching methods and non-technical skills within DevOps courses.

REFERENCES

[1] “Exploring the benefits of combining devops and agile,”F. Almeida et al., Future Internet, vol. 14, no. 2, p. 63, 2022.
[2] I. S. Committee et al., “Ieee standard for devops:building reliable and secure systems including application build, package, and deployment,” IEEE Std 2675-2021, pp. 1–91, 2021.
[3] G. Rong, S. Gu, H. Zhang, and D. Shao, “Devopsenvy: an education support system for devops,” in 2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T). IEEE, 2017, pp. 37– 46.
[4] M. Wiesche, “Are you ready for devops? required skill set for devops teams devops teams,” in Twenty-Sixth European Conference on Information Systems (ECIS2018), 2018, pp. 12–14.
[5] S. Ferino, M. Fernandes, A. Fernandes, U. Kulesza, E. Aranha, and C. Treude, “Analyzing devops teaching strategies: An initial study,” in Brazilian Symposium on Software Engineering, 2021, pp. 180–185.
[6] R. A. K. Jennings et al., “Devops-preparing students for professional practice,” in 2019 IEEE Frontiers in Education Conference (FIE). IEEE, 2019, pp. 1–5.
[7] E. Bobrov et al., “Teaching devops in academia and industry: reflections and vision,” in International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment. Springer, 2019, pp. 1–14
[9] H. B. Christensen, “Teaching devops and cloud computing using a cognitive apprenticeship and story-telling approach,” in Proceedings of the 2016 ACM conference on innovation and technology in computer science education, 2016, pp. 174–179.
[10] C. Jones, “A proposal for integrating devops into software engineering curricula,” in International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment. Springer, 2018, pp. 33–47.
[11] T. Kilamo et al., “Teaching collaborative software development: A case study,” in 2012 34th International Conference on Software Engineering (ICSE). IEEE, 2012, pp. 1165–1174.
[12] M. R. Marques et al., “A systematic mapping study on practical approaches to teaching software engineering,” in 2014 IEEE Frontiers in education conference (FIE) proceedings. IEEE, 2014, pp. 1–8.
[13] X. Bai et al.,“Thedevops lab platform for managing diversified projects in educating agile software engineering,” in Proceedings of the FIE 2018. IEEE, 2018, pp. 1–5.
[14] R. Chatley et al., “Lean learning-applying lean techniques to improve software engineering education,” in 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET). IEEE, 2017, pp. 117–126.
[15] A. M. Moreno et al., “Enriching traditional software engineering curricula with software project management knowledge,” in Proceedings of the 38th International Conference on Software Engineering Companion, 2016, pp. 404–411.

[16] R. Sadeghi et al., "Comparison of the effect of lecture and blended teaching methods on students' learning and satisfaction," Journal ofadvances in medical education & professionalism, vol. 2, no. 4, p. 146, 2014.

[17] H. Norouzi and S. Mohsenizadeh, "Jafary suny h, ebrahimzadeh s. the effect of teaching using a blend of collaborative and mastery oflearning models, on learning of vital signs: an experiment on nursing and operation room students of mashhad university of medical sciences,"Iranian Journal of Medical Education, vol. 11, no. 5, pp. 544–53, 2011.

[18] M. Kuhrmann, A practical approach to align research with master's level courses, in 2012, IEEE 15th International Conference on Computational Science and Engineering. IEEE, 2012, pp. 202–208.

[19] M. Fernandes, S. Ferino, A. Fernandes, U. Kulesza, E. Aranha, and C. Treude, "Devops education: An interview study of challenges andrecommendations," in 2022 IEEE/ACM 44th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET), 2022, pp. 90–101.

[20] A. Grotta and E. P. V. Prado, "Devops didactic transposition in is higher education: A systematic literature review," in Proceedings of theAmericas Conference on Information Systems: SIG ED - IS in Education, IS Curriculum, Education and Teaching Cases, 2022.

[21] L. Leite, C. Rocha, F. Kon, D. Milojicic, and P. Meirelles, A survey of devops concepts and challenges, ACM Computing Surveys (CSUR),vol. 52, no. 6, pp. 1–35, 2019. [34] P. S. Westwood, What teachers need to know about teaching methods, Aust Council for Ed Research, 2008.

[22] G. Bj¨orkStiernstr¨om, "Video tutorials for artists in the entertainment industry: A comparative study of what makes a video tutorial effective," 2017.

[23] Q.-x. Liu and J.-f. Shi, "An analysis of language teaching approaches and methods–effectiveness and weakness" Online Submission, vol. 4, no. 1, pp. 69–71, 2007.

[24] S. Ferino, M. Fernandes, E. Cirilo, L. Agnez, B. Batista, U. Kulesza, E. Aranha, and C. Treude. (2022) Overcoming challenges in devops education through teaching methods. [Online]. Available: https://doi.org/10.5281/zenodo.7135611

[25] M. Indiramma, "Project based learning—theoretical foundation of computation course," in 2014 International Conference on Interactive Collaborative Learning (ICL). IEEE, 2014, pp. 841–844.

[26] M. Pivec, O. Dziabenko, and I. Schinnerl, "Aspects of game-based learning," in 3rd International Conference on Knowledge Management, Graz, Austria, vol. 304, 2003.

[27] G. Akc¸ayır and M. Akc¸ayır, "The flipped classroom: A review of itsadvantages and challenges," Computers & Education, vol. 126, pp. 334–345, 2018.

[28] H. L. Zeng, D. X. Chen, Q. Li, and X. Y. Wang, "Effects of seminarteaching method versus lecture-based learning in medical education: ameta-analysis of randomized controlled trials," Medical Teacher, vol. 42,no. 12, pp. 1343–1349, 2020.

[29] B. O. Omatseye, "The discussion teaching method: An interactivestrategy in tertiary learning", Education, vol. 128, no. 1, 2007.

[30] O. Hazzan, N. Ragonis, and T. Lapidot, "Lab-based teaching." in Guideto Teaching Computer Science, Springer, 2020, pp. 221–249.

[31] V. M. Harder and G. L. Burns, "Geopardy-a computer-assisted reviewsession", Journal of Geological Education, vol. 40, no. 3, pp. 222–226,1992.

[32] I. Richardson and Y. Delaney, “Problem based learning in the softwareengineering classroom,” in 2009 22nd Conference on Software EngineeringEducation and Training. IEEE, 2009, pp. 174–181.
[33] Y. Akyuz, “Effects of intelligent tutoring systems (its) on personalizedlearning (pl),” Creative Education, vol. 11, no. 6, pp. 953–978, 2020.
[34] D. Kolb, Experiential learning: experience as the source of learning anddevelopment. Englewood Cliffs, NJ: Prentice Hall, 1984.
[35] F. Shahid, M. Aleem, M. A. Islam, M. A. Iqbal, M. M. Yousaf et al., “Areview of technological tools in teaching and learning computer science,”Eurasia Journal of Mathematics, Science and Technology Education,vol. 15, no. 11, p. em1773, 2019.
[36] A. J. Maletz, “Teaching in the dark: Increasing student engagementin comprehensive distance learning,” Master’s thesis, Western OregonUniversity, Monmouth, 2021.
[37] P. Salza, P. Musmarra, and F. Ferrucci, Agile Methodologies in Education: A Review. Singapore: Springer Singapore, 2019, pp. 25–45.