10-2021

Analyzing DevOps teaching strategies: An initial study

Samuel FERINO

Marcelo FERNANDES

Anny K. FERNANDES

Uirá KULESZA

Eduardo ARANHA

See next page for additional authors

Analyzing DevOps Teaching Strategies: An Initial Study

Samuel Ferino[1], Marcelo Fernandes[1,2], Anny Fernandes[1]
Uirá Kulesza[1], Eduardo Aranha[1], Christoph Treude[3]
[1]Federal University of Rio Grande do Norte (UFRN), Brazil,
[2]Federal Institute of Rio Grande do Norte (IFRN), Brazil, [3]University of Adelaide, Australia
samuellucas97@ufrn.edu.br,marcelo.fernandes@ifrn.edu.br,anny.klarice.fernandes.127@ufrn.edu.br
{uira,eduardoaranha}@dimap.ufrn.br,christoph.treude@adelaide.edu.au

ABSTRACT

DevOps refers to a set of practices that integrate software development and operations with the primary aim to enable the continuous delivery of high-quality software. DevOps has also promoted several challenges to software engineering teaching. In this paper, we present a preliminary study that analyzes existing teaching strategies reported in the literature. Our findings indicate a set of approaches highlighting the use of environments to support teaching. Our work also investigates how these environments can contribute to address existing challenges and recommendations of DevOps teaching.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; • **Software and its engineering** → **Software creation and management**.

KEYWORDS

DevOps, teaching practices, environment, challenges, recommendations

ACM Reference Format:
Samuel Ferino[1], Marcelo Fernandes[1,2], Anny Fernandes[1], Uirá Kulesza[1], Eduardo Aranha[1], Christoph Treude[3] . 2021. Analyzing DevOps Teaching Strategies: An Initial Study. In *Brazilian Symposium on Software Engineering (SBES '21), September 27-October 1, 2021, Joinville, Brazil.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3474624.3477071

1 INTRODUCTION

DevOps refers to a set of practices where software development, testing, and delivery happens with more speed, frequency, and reliability [11]. The software industry has high interest in those practices, because they enable enterprise competition through technology innovation in the provision of software services deployed in an agile approach. However, there is a deficit of qualified professionals to meet the demand [19]

DevOps teaching has its challenges [11]. Among them, DevOps increases the evaluation workload for educators [20], as well as requiring them to learn a set of tools to perform continuous delivery and integration pipelines [13]. From an educational point of view, there is also the difficulty of setting up the DevOps approach from scratch [7].

In this context, the study by Fernandes et al. [15] contributed with insights related to teaching DevOps. They provided a systematic literature review (SLR) of DevOps courses, presenting challenges and recommendations in teaching DevOps practices.

However, there has been limited research into effective teaching practices for incorporating continuous integration and delivery concepts into software engineering courses [13]. An obstacle when planning DevOps courses is that educators are not aware of available teaching practices [20]. Such practices contribute to the teacher's confidence in outlining the course to allow greater learning for the students.

Thus, it is important to develop mechanisms that act on the identification, consolidation, and sharing of available teaching practices. The goal of this study is to investigate the teaching practices used in existing DevOps courses and to identify how DevOps-supporting environments could contribute to teach DevOps. To achieve this goal, the following research questions (RQs) are answered:

RQ1: *What are the teaching practices used in DevOps courses?*
RQ2: *What are the challenges handled by the educational supporting environments to teach DevOps?*
RQ3: *What are the recommendations adopted by the educational supporting environments to teach DevOps?*

This study contributes by presenting the teaching practices for teaching DevOps cited in the literature selected in the systematic review by Fernandes et al. [15]. Hence, the results can allow a better elaboration of DevOps courses. A deeper analysis of the teaching practice of educational environments is also carried out. In addition, this work shows the state-of-the-art features that address specific DevOps educational challenges and recommendations pointed out by Fernandes et al. [15]. This study provides support to DevOps educators to improve their courses by presenting teaching practices for DevOps which are mentioned in the literature selected by Fernandes et al. [15]. Although we have used the same set of papers selected, our analysis and results are completely new and original.

The results of this work can benefit education professionals who work in DevOps teaching to reflect on the teaching practices adopted in the organization of their courses. Being informed about existing teaching strategies that are being adopted, educators could try to experiment with those strategies (or combinations of strategies) that they have not used before. The knowledge of a diversity of teaching practices can contribute to improving DevOps courses. Furthermore, this work can also motivate educators to explore or

SBES '21, September 27-October 1, 2021, Joinville, Brazil

ACM ISBN 978-1-4503-9061-3/21/09…$15.00
https://doi.org/10.1145/3474624.3477071

combine new educational strategies not applied before. In addition, our study also indicates how existing DevOps-Supporting Environments can be improved to better address existing challenges and recommendations for DevOps education.

The remainder of this paper is structured as follows. We review related work in Section 2. Section 3 identifies our research questions and our study procedures. We report and discuss teaching practices findings, especially about DevOps-supporting environments in Sections 4 and 5. Section 6 reviews the threats to validity of this study. Finally, we make final remarks of our work and discuss research opportunities in Section 7.

2 RELATED WORK

This paper addresses the usage of teaching practices in DevOps education, with a particular focus on educational DevOps-supporting environments use.

Fernandes et al. [15] provides a Systematic Literature Review (SLR) about challenges and recommendations in DevOps Education. It was not studied teaching practices but they identified 73 challenges and 85 recommendations grouped in 7 themes, of which 26 challenges and 23 recommendations are DevOps specific. We used this SLR and their primary papers, challenges, and recommendations in our research as a source of information to investigate teaching practice approaches and DevOps-supporting environments in particular.

In this context, a challenge and a recommendation do not necessarily have a teaching practice related to them. We can show a recommendation's example with teaching practice included as "project-based teaching has proved to be successful as a knowledge transfer mechanism". In this case, the educational strategy is project-based learning. But we also can find a recommendation without a teaching practice such as "DevOps should be a proper course in the curriculum".

Rong et al. [26] analyzed DevOps-supporting environments to propose the DevOpsEnvy tool. The authors used Christensen's case study paper [11] as the source of challenges handled by supporting environments. Our work does not offer a specific tool; it analyses all available academic tools using broader challenges found by Fernandes et al.'s SLR [15]. Moreover, we also used recommendations identified in the same SRL to guide our analysis.

Pang et al. [25] investigated the DevOps education area, from academic and industrial perspectives. However, they did not analyze teaching practices used for educators nor study existing DevOps supporting environments.

3 METHODOLOGY

This section presents the methodology used in this work. First, Section 3.1 introduces the definitions of this study. Then, Section 3.2 outlines the procedures applied during this research.

3.1 Study definitions

This study uses the following definitions. An *educational supporting environment* is a tool or platform that is used to support teaching activities. A *challenge* is a problem that brings difficulties to a DevOps course preparation or execution. Finally, a *recommendation* is a real and specific proposition to address a problem, making the learning process easier [15].

By answering RQ1, we aim to identify the usage of teaching practices in academic DevOps education experience reports. RQ2 guides the investigation of how supporting environments address existing DevOps education challenges. Finally, the answer to RQ3 highlights the DevOps recommendations adopted by those environments.

3.2 Study Procedures

The study followed the procedures described in this section. First, all 18 papers identified by the systematic literature review of Fernandes et al. [15] were read by two reviewers who collected the data needed to answer RQ1, RQ2 and RQ3. We followed this process since the existing literature review searched for all papers containing experience reports in DevOps education. These experience reports are likely to show education strategies in their execution. When necessary, the consistency of the collected data is discussed by the reviewers and the other authors of this work.

To answer RQ1, the reading of the 18 primary studies focuses on indications of teaching practices. For instance, Christensen [12] contains the statement *Crunch is a batch program which runs at scheduled intervals, and performs an assessment of all submitted exercises for all groups* which indicates the use of an educational environment called Crunch. Another example is Capozucca et al. [9], which states that *The course was designed as a series of [...] practical sessions*, indicating the use of labs. These results help to understand which papers report on the use of teaching practices and which ones only mention the usage possibility without the application evidence, as we can see in *DevOps is the ideal candidate for non-traditional ways of teaching, such as [...] flipped classroom* referencing to the flipped classroom.

Otherwise, the paper will be classified as non-identified usage if there is no mention of usage. When many teaching practices are identified in a study, this will be considered as a combination by default, except if the paper contains explicit indications to the contrary.

To answer RQ2, the reviewers looked for indicators of educational environment features that addressed one of the 26 specific DevOps challenges identified in the literature review. For example, for the specific challenge *setup the tools and workflows increases the effort for educators*, the feature quick setup (*... The main goal of CDEP is to provide a [...] quick to setup teaching tool*) is identified and related to the CDEP educational environment [26]. Thus, it is understood that CDEP addresses the setup challenge.

Concerning RQ3, reviewers looked for educational supporting environments incorporating one of the 23 specific DevOps recommendations identified in the literature review. For example, for the recommendation *without our automated deployment pipeline the customer feedback would not have been possible in such an easy way*, the statement *... a collaborative development platform by integrating GitLab with plug-in tools for continuous [...] deployment* is related to the educational environment DevOps Lab Platforms which concerns the functionality of continuous deployment and deployment automation. As such, it is understood that DevOps Lab Platform adopts this recommendation.

4 RESULTS

In this section, we present the obtained answers to the research questions of this preliminary study. We provide spreadsheet [14] containing all data collected about DevOps teaching strategies worked in this paper.

4.1 RQ1: What are the teaching practices used in DevOps courses?

Figure 1 shows the distribution of the 13 teaching practices that are reported in the literature. We can observe that the three most cited strategies are educational supporting environment (7 paper citations), lectures (7 citations), and laboratories (6 citations). On the other hand, the three least cited strategies are studio-based learning, experiential learning and case study with just one paper citation each.

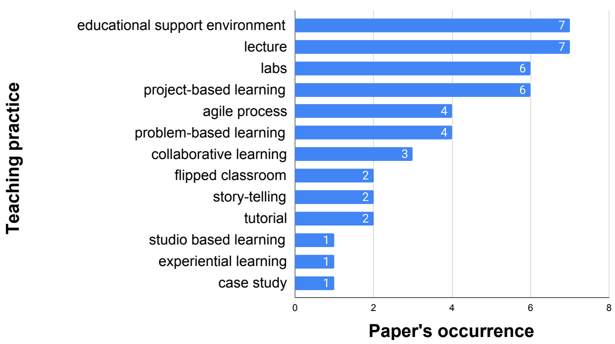

Figure 1: Teaching Practices vs Paper's occurrence

The supporting educational environments are used to support teaching by facilitating the course assessment and promoting the integration of tools. These environments are more discussed in Sections 4.2 and 4.3.

The lecture strategy is a strategy used in several courses. It is commonly used to introduce new content. Airaj [1] adopted it during the first part of their course. Krusche and Alperowitz [19] prepared lectures to teach the concepts of meeting management and agile methods. Our analysis also identified the report of the integrated use of lectures with problem-based learning and agile processes [7].

The use of Labs was also identified as an teaching practice. Capozucca et al. [9] have combined and interchanged the use of lectures with practical sessions in labs. Eddy et al. [13] report multiple benefits of the use of labs to teach continuous integration and delivery. Finally, Benni et al. [6] mention the integration of labs and case study strategies as a way to improve DevOps teaching.

Project-based learning is an teaching practice that implies collaboration between two or more students when planning, implementing, and/or evaluating a course, which mainly involves the exchange of training expertise and reflective conversation [3]. The project-based learning strategy was reported to support the organization of some DevOps courses [7] [9] [19] and was additionally cited in [8] [20]. Problem-based learning provides a powerful alternative to passive lectures by introducing concepts and motivating learning in an active and cooperative environment using real-world problems [2].

The adoption of agile processes is also considered as a teaching strategy in course project design. It encourages communication and collaboration among team members [5]. By means of checkpoints, the students gradually deliver improved versions of their projects along the course timeline [7] [9].

Another teaching practice is collaborative learning. In peer instruction, the instructor poses a question with discrete options and gives students the chance to consider and record their answers individually, often by voting using clickers. Students then discuss their answers with neighbors, explaining their reasoning, before being given a chance to vote again. Finally, the instructor discusses the answer to the question, often soliciting input from the class [18]. Collaborative learning has also been used in peer reviewing activities to distribute points depending on the review quality [9].

The flipped classroom model is based on the idea that students read course literature and assimilate lecture material through videos at home and engage in teacher-guided problem-solving, analysis and discussions in class [22]. It was mentioned in two papers [8] [17]. Flipped classroom has also been used as a supplementary teaching practice to the use of labs [17].

Similarly, tutorials is also mentioned as a supplementary teaching practice. Installation and configuration guides of modern tools such as Jenkins are mentioned as examples of such tutorials [16].

Finally, we also identified mentions of other teaching practices such as studio-based learning and experiential leaning. Studio-based learning is a pedagogical technique that promotes learning through the iterative construction and review of problem solution [10]. Experiential learning is a modern teaching practice based on the understanding of different ways of students' learning [27].

Table 1 shows the distribution of teaching practices by papers with effective use. In several papers [24] [23] [26], we identified educational supporting environment as unique teaching practice. Lecture is always used with other existing teaching practices. Capozucca et al [9] is the research work that combines the biggest number of teaching practices: lecture, project-based learning, agile process, labs, and collaborative learning.

Table 2 shows cited teaching practices distributed by papers. These practices have only been discussed as possibilities in the respective publications without identified use. Studio based learning and experiential learning do not have their use identified.

4.2 RQ2: What are the challenges handled by the educational supporting environments to teach DevOps?

We identified a total of five different supporting environments in our analysis: DevOps Lab Platform [4] [5], ALECSS [24] [23], Crunch [12], DevOpsEnvy [26], and CDEP [13]. Table 3 shows each environment by highlighting their main features.

The DevOps Lab Platform enviroment [5] [4] is an online platform that was developed based on GitLab but that also integrates build management (Jenkins), repository analysis (Codeface) and code quality analysis (SonarQube). It automatically delivers customized reports related to code smells, test coverage, bug reports, task allocation, scheduling, team member contribution, frequency

Table 1: Used Teaching Practices

	Paper															
Teaching Practice	[24]	[23]	[26]	[4]	[5]	[12]	[11]	[13]	[1]	[21]	[7]	[9]	[16]	[19]	[6]	[17]
educational support environment	X	X	X	X	X	X		X								
agile process				X	X						X	X				
story-telling						X	X									
labs								X		X		X	X		X	X
lecture									X	X	X	X	X	X	X	
project-based learning									X		X	X		X		
problem-based learning							X				X					
tutorial													X	X		
case study															X	
collaborative learning												X				
flipped classroom																X

Table 2: Non-Identified Usage of Teaching Practices

	Paper	
Teaching Practices	[20]	[8]
project-based learning	X	X
problem-based learning	X	X
studio based learning	X	
collaborative learning	X	X
experiential learning	X	
flipped classroom		X

Table 3: Main Features of the DevOps-Supporting Environments

DevOps-Supporting Environment	Main Features
DevOps Lab Platform	automated assessment; integrating various tools; evaluate team workload
ALECSS	automated assessment; integrating various tools
Crunch	automated assessment
DevOpsEnvy	automated assessment; integrating various tools; collaborating on projects easily
CDEP	simple design that facilitates the learning curve; integrating various tools

and adequacy of commitments to gradually improve student's skills. The Crunch tool [12] is a batch program that can be executed periodically to evaluate a set of exercises submitted by students.

The ALECSS educational supporting environment [23] [24] integrates existing build management (Jenkins), code versioning (Git), unit testing (JUnit), and code quality (Checkstyle, FindBugs) tools. ALECSS also provides support to monitor the progress of students' activities and to give hints to them related to the continuous integration process.

The DevOpsEnvy [26] educational environment integrates existing tools such as code versioning (Git), continuous integration and delivery (Jenkins), code quality (SonarQube), build management (Maven and Gradle) and deployment (Docker). It also allows managing and monitoring students that are working on DevOps practices. Finally, the CDEP [13] environment allows students to visualize and exercise projects that are being delivered and deployed using Jenkins and Docker tools.

Table 4 shows how the different functionalities provided by the environments address the specific DevOps challenges.

The *DevOps increases evaluation workload for teachers* challenge, for example, belongs to the theme of evaluation. All educational environments except CDEP work to address this challenge through automation in assessment. Crunch enables the evaluation of performed exercises. DevOps Lab Platforms allows generating reports that can provide feedback to students. DevOpsEnvy allows teachers to automatically assess student progress. ALECSS supports evaluation from messages generated by the Checkstyle and FindBugs tools.

The *Setup the tools and workflows increase the effort for educators* challenge belongs to the theme of environment setup. All educational environments address this challenge (except Crunch) through the integration of DevOps tools. Students have only to register their information in Jenkins and SonarQube to configure DevOpsEnvy.

In addition, DevOpsEnvy also deals with the *teach DevOps mindset/cultural* challenge of the theme DevOps concepts by enabling the collaboration of students in the project developed by their groups. This is supported by the team members management functionality. The DevOps Lab Platform also addresses the specific challenge *Teaching DevOps mindset is difficult to assess* of the evaluation theme

Table 4: Challenges handled by functionalities

Functionality	Specific challenge
Automated assessment *<DevOpsEnvy>* *<Crunch>* *<DevOps Lab Platform>* *<ALECSS>*	*DevOps increases evaluation workload for teachers*
Integrating various tools *<DevOpsEnvy>* *<CDEP>* *<DevOps Lab Platform>* *<ALECSS>*	*Setup the tools and workflows increases the effort for educators*
Evaluate team workload *<DevOps Lab Platform>*	*Teaching DevOps mindset is difficult to assess*
Simple design that facilitates the learning curve *<CDEP>*	*DevOps tools are complex*
Collaborating on projects easily *<DevOpsEnvy>*	*Teach DevOps mindset / cultural*

by defining metrics to assess the efforts of each student to collaborate in a project developed with other students. CDEP also handles the specific challenge *DevOps tools are complex* through the environment design which has a low learning curve.

4.3 RQ3: What are the recommendations adopted by the supporting environment educational to teach DevOps?

Table 5 shows that the six identified functionalities from the educational environments address three specific recommendations distributed across two themes. DevOps Lab Platform is the only educational environment that contributes to two functionalities that adopt specific recommendations. The other educational environments only address or implement a single specific recommendation.

The following recommendation *Our viable design choices for automated assessment: external test case deployment using container technology, validation through user interface input/output, service configuration through chained property files, and fine-grained inspection through a special logging mechanism* is adopted by most of the educational environments. All of them except CDEP have adopted this recommendation. DevOpsEnvy, DevOps Lab Platform, and Crunch used the deployment of artifacts through containerization (Docker) as a form of testing in this context. ALECSS adopts the strategy of collecting logs of student activities to assess their progress.

The DevOps Lab Platform also addresses the specific recommendation *without our automated deployment pipeline the customer feedback would not have been possible in such an easy way* through the continuous deployment implementation. CDEP adopts the specific recommendation *The pipeline used was designed to help instructors introduce continuous integration and delivery into preexisting courses and allow students to visually understand the processes of continuous delivery and continuous integration* using the continuous delivery pipeline implementation.

Table 5: Recommendations adopted by functionalities

Functionality	Specific recommendation
Deployment using container technology; Collect student's activity log;	*Our viable design choices for automated assessment: external test case deployment using container technology, validation through user interface input/output, service configuration through chained property files, and finally fine-grained inspection through a special logging mechanism*
Teach concepts using developed pipeline	*The pipeline used was designed to help instructors introduce continuous integration and delivery into preexisting courses and allow students to visually understand the processes of continuous delivery and continuous integration*
Continuous deployment implementation	*Without our automated deployment pipeline the customer feedback would not have been possible in such an easy way*

5 DISCUSSION

Many different teaching practices are used in DevOps Education, according to Figure 1. However, unusual strategies such as collaborative learning have few citations. Some of citations reveal that strategies like flipped classroom have complementary usage in classes [17]. Other strategies like experimental learning and studio-based learning are just cited and not used effectively [20]. This shows research opportunities to discover best practices to implement not validated teaching practices in DevOps Education.

A more significant part of primary papers from Fernandes et al.'s SLR [15] uses combinations of teaching practices in DevOps Education. As an example, Capozucca et al. [9] used five teaching practices in their course. On the other hand, some papers just cited varieties without applying in a DevOps class. Moreover, there are still many associations that have not been tried yet. This situation opens up research opportunities to validate best practices to combine teaching practices.

All educational environments with the exception of Crunch [12] offer an integrated and automatic setup to run continuous integration (CI). The majority of them also provide continuous delivery (CD). This is achieved by connecting many different tools and services. This feature proved to be very useful because significant industrial DevOps environments like Github and Gitlab evolved to offer this internally with Github Actions and Gitlab CI/CD services, respectively. Nowadays, it is possible to use a single service to run CI and CD.

Educational DevOps-supporting environments tackled 5 of the 26 specific challenges in DevOps Education. 73% of the tool features targeted only two specific challenges. Moreover, these environments addressed 3 of 23 specific recommendations in DevOps Education, and 67% of the tool features targeted only a single recommendation. Current DevOps-supporting tools do not address the majority of relevant challenges and recommendations in DevOps Education. For example, the DevOps Concepts challenge *CD and related concepts are complex* and the recommendation *the course must focus on the concepts, and use tools only as an illustration* are significant ones which remain to be addressed.

We observed that all environment functionalities are related to recommendations from the topic of assessment or DevOps concepts, showing a good potential to address other topics such as pedagogy. For instance, supporting environment can apply the specific recommendation *To simulate real-life software engineering, the labs must be define an open and informal specification expressed in business terms, and it will be up to the students to design the right architecture, implement it in an iterative way and support its deployment through of a continuous delivery pipeline.* from pedagogy theme inserting a business scope.

Finally, untackled specific DevOps educational challenges and recommendations could guide the requirements of the new generation of Educational DevOps-supporting environments.

6 THREATS TO VALIDITY

Any secondary study has common threats to validity related to the selection of primary studies. In this study, this selection is based on the results of a systematic literature review [15] published in 2020. In this way, it is possible to have more recent papers with information not considered by this research. However, the findings presented here are still valid and can be complemented in a future study update.

Another common threat to validity is research bias. However, this study considered primary studies found in an existing systematic literature review [15]. Also, Section 3 describes the followed methodology to understand and identify the teaching strategies. Two researchers worked on the data collection. Periodically, they discussed the results, and the other PhD researchers continuously reviewed the obtained results.

7 FINAL REMARKS

This paper presented a set of teaching practices used in DevOps Education and showed how researchers use combinations of these practices. This work also investigated DevOps-supporting teaching environments, showing how these tools deals with previously identified DevOps education challenges and how they adopt teaching recommendations. The majority of challenges and recommendations are still not addressed.

In this context, as future work, new studies can propose improvements in these educational tools to include more recommendations and address more challenges. Also, it would be helpful to compare the teaching practices found in these publication with the ones used in the classroom.

Acknowledgements. This work is partially supported by INES (www.ines.org.br), CNPq grant 465614/2014-0, CAPES grant 88887. 136410/2017-00, FACEPE grants APQ-0399-1.03/17, PRONEX APQ/ 0388-1.03/14, and IFRN.

REFERENCES

[1] Mohammed Airaj. 2017. Enable cloud DevOps approach for industry and higher education. *Concurrency and Computation: Practice and Experience* 29, 5 (2017), e3937.
[2] D. Allen et al. 1996. The power of problem-based learning in teaching introductory science courses. *New directions for teaching and learning* 1996, 68 (1996), 43–52.
[3] M. Almulla. 2020. The Effectiveness of the Project-Based Learning (PBL) Approach as a Way to Engage Students in Learning. *Sage Open* 10, 3 (2020), 2158244020938702.
[4] X. Bai et al. 2018. Continuous delivery of personalized assessment and feedback in agile software engineering projects. In *Proceedings of the ICSE 2018.* 58–67.
[5] X. Bai et al. 2018. The DevOps Lab Platform for Managing Diversified Projects in Educating Agile Software Engineering. In *Proceedings of the FIE 2018.* IEEE, 1–5.
[6] B. Benni et al. 2018. Teaching DevOps at the Graduate Level. In *Proceedings of the first international workshop devops 2018.* Springer, 60–72.
[7] E. Bobrov et al. 2019. Teaching DevOps in academia and industry: reflections and vision. In *Proceedings of the first international workshop DevOps 2018.* Springer, 1–14.
[8] J.M. Bruel et al. 2018. DevOps' 18 Education Panel. In *Proceedings of the first international workshop devops 2018.* Springer, 221–226.
[9] A. Capozucca et al. 2018. Design of a (yet another?) DevOps course. In *Proceedings of the first international workshop devops 2018.* Springer, 1–18.
[10] A. Carter et al. 2011. A review of studio-based learning in computer science. *Journal of Computing Sciences in Colleges* 27, 1 (2011), 105–111.
[11] H. Christensen. 2016. Teaching DevOps and cloud computing using a cognitive apprenticeship and story-telling approach. In *Proceedings of the ITiCSE 2016.* 174–179.
[12] H. Christensen. 2018. Crunch: Automated Assessment of Microservice Architecture Assignments with Formative Feedback. In *Proceedings of the ECSA 2018.* Springer, 175–190.
[13] B. Eddy et al. 2017. A pilot study on introducing continuous integration and delivery into undergraduate software engineering courses. In *Proceedings of the CSEE&T 2017.* IEEE, 47–56.
[14] S. Ferino et al. 2021. *Research Artifacts.* https://zenodo.org/record/5132912#.YPxcVzrQ9E4
[15] M. Fernandes et al. 2020. Challenges and Recommendations in DevOps Education: A Systematic Literature Review. In *Proceedings of the SBES 2020.* 648–657.
[16] L. Greising et al. 2018. Introducing a Deployment Pipeline for Continuous Delivery in a Software Architecture Course. In *Proceedings of the ECSEE 2018.* 102–107.
[17] R. Jennings et al. 2019. DevOps-Preparing Students for Professional Practice. In *Proceedings of the FIE 2019.* IEEE, 1–5.
[18] J. Knight et al. 2018. Peer instruction. *CBE—Life Sciences Education* 17, 2 (2018), fe5.
[19] S. Krusche et al. 2014. Introduction of continuous delivery in multi-customer project courses. In *Proceedings of the ICSE 2014.* 335–343.
[20] K. Kuusinen et al. 2019. Industry-academy collaboration in teaching DevOps and continuous delivery to software engineering students: towards improved industrial relevance in higher education. In *Proceedings of ICSE 2019.* IEEE, 23–27.
[21] M. Mazzara et al. 2018. Teaching DevOps in corporate environments. In *Proceedings of the first international workshop devops 2018.* Springer, 100–111.
[22] J. Nouri. 2016. The flipped classroom: for active, effective and increased learning–especially for low achievers. *International Journal of Educational Technology in Higher Education* 13, 1 (2016), 1–10.
[23] M. Ohtsuki et al. 2016. Software engineer education support system ALECSS utilizing DevOps tools. In *Proceedings of the iiWAS 2016.* 209–213.
[24] M. Ohtsuki et al. 2019. Utilizing Software Engineering Education Support System ALECSS at an Actual Software Development Experiment: A Case Study. In *Proceedings of the CSEDU 2019.* 367–375.
[25] C. Pang et al. 2020. Understanding DevOps Education with Grounded Theory. In *Proceedings of the ICSE 2020.* IEEE, 1–12.
[26] G. Rong et al. 2017. DevOpsEnvy: an education support system for DevOps. In *Proceedings of the CSEE&T 2017.* IEEE, 37–46.
[27] V. Sharlanova. 2004. Experiential learning. *Trakia Journal of Sciences* 2, 4 (2004), 36–39.