Original article

EMBEDDING THE PRINCIPLES AND PRACTICES OF DEVOPS INTO THE IT EDUCATION

Teklehaimanot Yemane[1], Jelena Mihajlović Milićević[1], Miloš Radenković[2]

[1]Faculty of Organizational Sciences, University of Belgrade, Serbia
[2]School of Computing, Union university, Serbia

Received: 30 April 2021
Revised: 5 May 2021
Accepted: 10 May 2021

Abstract: With the rapid advancements of Information Technology (IT) in enterprises, the existing IT educational methods in universities are struggling to meet the new functional and technical requirements. This paper offers an overview of DevOps principles and practices and proposes an effective method for augmenting IT education with these concepts. To achieve this, a conceptual framework needs to be developed where the principles and practice of DevOps can be easily embedded into existing IT education courses. In addition to this framework, an end-to-end pipeline and toolchains for efficient execution of selected IT education programs need to be developed. These pipelines must support the working culture and enhance the productivity of students, professors, and teaching boards while still being in line with the requirements of enterprises. The proposed model is developed from six aspects of DevOps which are continuous development, continuous integration, continuous testing, packaging and staging, release automation, and performance/relevance monitoring. Through the embedding of DevOps culture and practices into existing IT education, we expect to advance the relevance of the education-industry flow while providing a platform for continual learning and experimentation. The proposed framework was implemented and evaluated at two IT courses at the Faculty of Organizational Sciences, University of Belgrade.

Keywords: DevOps, IT education, industry relevance, IT courses, continuous development, continuous integration, continuous testing, packaging and staging, release automation, performance monitoring.

1. INTRODUCTION

The existing IT education methods in universities are often not following in step with the rapidly growing new functional and technical requirements in enterprises (Kuusinen, & Albertsen, 2019). DevOps, short for development and operations is a software development method that extends the agile philosophy to rapidly produce software products and services while improving operations performance and quality assurance (Leite et al., 2019; Bass et al., 2015; Bayser et al., 2015; Riungu-Kalliosaari et al., 2016). DevOps is defined as a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production while ensuring a high quality of code and its deployment which is required by today’s enterprises (Bruel, & Jiménez, 2019; Ohtsuki, & Kakeshita, 2019; Ohtsuki et al., 2016). By embedding the culture, principles, and practices of DevOps into the existing IT education, we expect to advance the

Corresponding author. Email: jelenamm@elab.rs
ISSN 2560-4961 (online)

https://doi.org/10.18485/epmj.2021.11.2.1

relevance of the education-industry flow, feedback, continual learning, and experimentation (Bobrov et al., 2020; Yang et al., 2020). As stated in Hills (2020), adopting DevOps practices often requires both technical and cultural changes within an institution, as the classic and isolated teams are replaced by cross-functional teams. DevOps mainly depends on toolchains and automation of practices like testing, build, continuous integration, continuous delivery, and continuous deployment (Kim et al., 2016).

As we experience the rapid developments of Information Technology (IT) in enterprises, the practical relevance of the education systems in universities to the technological requirements of enterprises has greatly changed. By employing the principles and best practices of DevOps in IT education, this difference can be greatly balanced. In this study, we introduce a new pedagogical approach for a modern and comprehensive IT education based on the principles, concepts, techniques, and toolchains of DevOps.

The designed series of comprehensive lab activities and templates are given in selected IT education courses. These templates guide students in developing end-to-end DevOps solutions that include virtualization, containerization, continuous development, a configuration of continuous integration, continuous testing, packaging and staging, performance monitoring, constructing a build pipeline, and automating the deployment of new versions of software systems or products upon change.

The remainder of this paper is organized as follows. In Section II, we provide an overview of existing efforts of enriching IT education courses with DevOps practices while making use of professional and enterprise tools and techniques. The literature review is briefly mentioned in Section II. Section III explains the proposed DevOps lab activity templates created for this study. Section IV will offer a short evaluation of both the student and professor's experiences during these activities. Section V offers a brief conclusion in which implications of the study are considered as well as the potential for future research.

2. LITERATURE REVIEW: DevOps in IT education

As an evolving method that supports the fast delivery of software products, services, and features with reliable quality, DevOps is attracting more and more industries, where it has become a mainstream approach for software development and operations. In line with this, there are several initiatives in introducing DevOps concepts into the education sector in several universities which are offered to students majoring in computer science and software engineering.

Chatley et al. (Chatley, & Procaccini, 2020) have applied DevOps practices in their Software Engineering Programme, through many different projects as an underpinning thread. Unlike other practices where the DevOps skills are isolated from the regular IT skills, the authors preferred to present the DevOps-related skills by integrating them into existing core courses. To accomplish this, the authors' set up what they called a DevOps Lab. The main idea of the DevOps Lab exercise focused on setting up a pipeline for the building, testing, and deployment of a Java web application provided by the authors in a Git repository. The students would iteratively add features with small code changes by following the best practices of DevOps pipelines. However, the DevOps lab takes only two weeks, and students are required to study a variety of documentation to work on the lab exercises over a very short period. As a result, the DevOps lab focuses only on the automation part of the DevOps without dealing with the culture/process aspects.

Similarly, Mark Hills (2020) introduced DevOps into a semester-long elective software development course in a graduate software engineering program. As described by the author, a collection of DevOps assignments were given, including creating Docker containers, configuring continuous integration services, constructing a build pipeline, and automating deployment. The author briefly analyzed the performance of students on these exercises and identified areas where support was needed. The students used various tools such as Apache Maven, JUnit, Travis CI, FindBugs, Github, Docker images, and Google

Kubernetes Engine. The final goal is the integration of GitHub and the Google Cloud Platform, to automate the build and deployment process, allowing new images to be created and deployed when changes are pushed to GitHub. The efforts of the authors pose an excellent all-around approach to introducing DevOps into IT Education, but students report that they were overwhelmed with the number of new tools and were not very productive as a result.

A comprehensive study similar in scope to the one presented in this paper was conducted by Kati Kuusinen and Albertsen (2019). The objective of the study was to investigate how university courses requiring multidisciplinary lecturer skills and complex execution architectures can be organized in industry-academia collaboration to improve the industrial relevance of higher education. To achieve this the authors designed and executed a stand-alone course that includes concepts of Continuous Delivery and DevOps organized in company-university collaboration. In addition, the authors developed learning objectives with the following question in mind: What learning objectives promote both learning of underpinning theories and abstraction frameworks as well as learning of practical skills that improve the employability of new software engineering graduates. The authors outlined nine learning outcomes that can be classified into three groups: cognitive competencies (or knowledge), functional competencies (or skills), and social and meta-competencies (or competencies).

In the research we conduct, the gap is finding an effective and feasible way to adapt university courses with the technologies and knowledge used in industry, while providing an environment conducive to education for continuous learning and experimentation. This is planned to be achieved by incorporating DevOps culture and practice, as well as agile methodologies into existing IT education, and the expected result would be a significant change in work culture and increased productivity of students, professors, and teaching boards, all in line with company requirements.

3. METHODOLOGY

3.1. Research goals

The main goal of this research is to develop a conceptual framework where the principles and practices of DevOps can be adequately embedded into the existing IT education courses. In addition to the framework, an end-to-end pipeline was developed to ensure effective execution of selected IT education programs, where the productivity of students, professors, and teaching boards are enhanced to be in line with the requirements of enterprises. To apply the framework and the pipeline in practice, we have formed a lab course called *“Educational DevOps IoT Lab”*. Through such a lab, DevOps can be implemented into existing curricula while being based on the infrastructure for education, monitoring, and evaluation of student projects.

3.2. Procedures

The model contains the following elements:

- Students educations about DevOps principles and practices through IT courses,
- IoT platform and Educational DevOps IoT Lab design and implementation.

3.2.1. DevOps Principles and Practices

DevOps is a collaborative and multidisciplinary organizational effort to automate the continuous delivery of new software updates while guaranteeing their correctness and reliability (Leite et al., 2019). DevOps also promotes fast and continuous feedback from operations to development to detect problems before customers are impacted (Lopez-Pena et al., 2020). Desiring to improve their delivery process, many enterprises are widely adopting DevOps in their software development ecosystem.

In this section, we describe how we have designed a DevOps IoT Lab ecosystem to introduce students to the DevOps practices, by embedding the DevOps tools and principles within an existing IoT Course.

For students to understand and work effectively in a DevOps environment, they must learn how to:

a) break down their projects into thin chunks that can be developed independently,
b) design and implement automated tests,
c) configure and maintain production environments and perform the IT operations tasks,
d) Select and configure suitable tools.

In short, the objective of DevOps is to minimize the time interval between two commits, so that the new version can be integrated, tested, and deployed to end-users timely.

3.2.2. IoT platform and Educational DevOps IoT Lab design

Internet-of-Things (IoT) applications are being developed for numerous domains including industry, healthcare and well-being, transportation, entertainment, etc., and deployed in smart environments (Atzori et al., 2017). Applications of IoT in those environments integrate sensors, actuators, computing resources, and communication components to control physical processes and monitor critical tasks.

Typical IoT system includes four components, figure 1:

a) **Hardware** - a set of connected physical devices, such as servers, physical objects, sensors, actuators, etc.
b) **Communication networks** – which interconnect other components
c) **Software infrastructure** - the platform for development, deployment, and execution of IoT applications
d) **Applications** – that offer vertical solutions over those infrastructures.

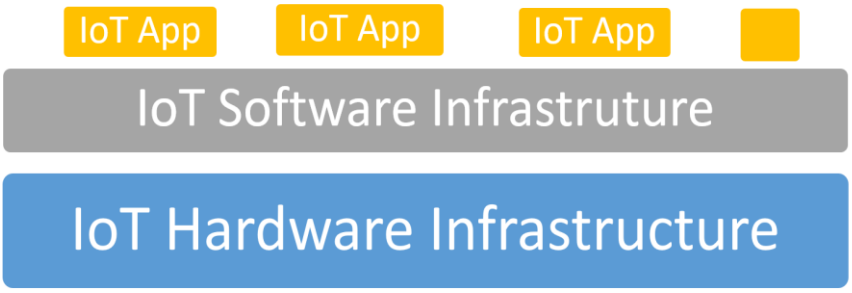

Figure 1: IoT Environment Components

However, for educational purposes, to present all those components might be very complex, difficult for students as well as expensive for educational institutions. In this study we design a simplified lab environment, affordable and easy to follow which is presented in three phases:

- **Phase 1 – Understanding IoT Development Tools and Working on DevOps Environment**: In this phase, individual students learn how to use simplified IoT development tool for prototyping and/or production purposes and practically integrate IoT development tools with DevOps environment to learn DevOps tools, principles, and practices including but not limited to *version control*, *build stage*, and *deploy stage*. The objective of this lab exercise is for students to be able to:
    - Use git source control and integrate node-RED with GitHub
    - Package IoT apps (node-RED apps) in containers using Docker containerization tool
    - Manage application configuration details for external services such as a database or messaging service.

The duration of the Lab exercise is one to three days, depending on the schedule of the IoT course, and individual student’s competency. Students will be given orientation and a brief introduction of the necessary tools with a demonstration. Afterward, students are given a small exercise to develop and test IoT apps in a virtual setting.

- **Phase 2 (optional) – Implementing DevOps IoT Infrastructure for Educational Purpose:** In this phase, students are organized in teams and will be asked to make a comparative study on advanced DevOps tools and provide an implementation design for educational settings. This phase is optional in that it could also be done in close collaboration with the IoT instructors, teaching assistants, and the research team of this study. The objective of this phase is to design, test, and implement an end-to-end DevOps IoT Infrastructure for teaching purposes by following the DevOps principles and practices and selecting from a variety of available tools such as Jenkins, Travis, Tekton, GitHub, Kubernetes, cloud infrastructure services and preferred IoT Platforms.

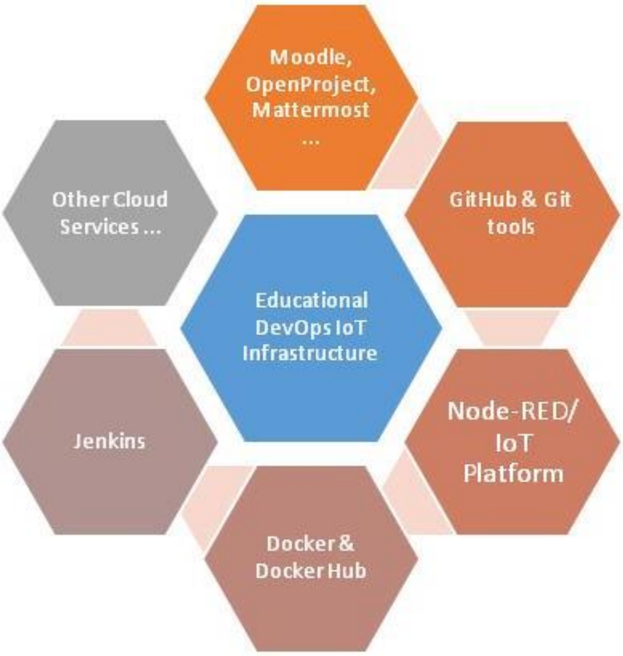

Figure 2: Educational DevOps IoT Infrastructure - A high-level design

Besides the Educational DevOps IoT Infrastructure will make use of existing e-learning systems, team productivity, and project management tools in a virtual setting as illustrated in figure 2. As a result of this phase, students are provided with a reliable DevOps IoT Infrastructure to work on advanced IoT application projects in the third phase. They learn DevOps principles and practices such as building, testing, continuous integration, continuous deployment, and continuous monitoring in an automated environment.

- **Phase 3 – Developing IoT Application within the Educational DevOps IoT Infrastructure:** After the Educational DevOps IoT Environment is set, students organized in teams will be given a set of IoT application projects that will be evaluated in terms of the usage of DevOps tools, principles, and practices and application functionality.

3.3. Research context and implementation details

To be suitable for educational purposes, the platform needs to offer the ability to monitor a large number of IoT projects in addition to offering well-designed project management functionalities. The *Educational DevOps IoT Lab* approach uses DevOps tools and SAFe agile framework to better adapt itself to IoT-based projects, smart environments, and working in virtual teams.

The research was conducted and evaluated at the Department of Electronic Business of the Faculty of Organizational Sciences. The goal was to develop a suitable system for organizing the workload of virtual teams, on individual IoT projects. The SAFe framework and the *Educational DevOps IoT Lab* approach were used. The research was conducted within undergraduate studies in two subjects: Internet of things and Risk Management in e-business. In both courses, students had lectures and exercises on agile SAFe and Scrum methodologies, the *Educational DevOps IoT Lab* approach as part of the SAFe framework, and the OpenProject project monitoring tool. Students who attended Risk Management in e-business were given the task of managing IoT projects and had the role of SAFe Scrum master. IoT teams were composed of students who shared the roles of programmer, designer, and tester. Each team had 3 to 5 members. (Mihajlović Milićević et al., 2019)

Professors had the role of product owners and they created Product Backlogs, a list of features that were assigned to each virtual team. Team members and the SAFe Scrum Master received clearly defined tasks from the Product Owner. Students were free to organize the work dynamics and to choose communication tools. SAFe Scrum masters were assigned to keep the team focused, make sure the team was working, and communicate with the product owners (Royle & Nikolic, 2016). Team members partly used the *Educational DevOps IoT Lab* approach in the development of the IoT project. Code versions were stored on GitHub and documentation on the Wiki. Both of these tools were available to all team members, SAFe Scrum master as well as Product Owners through the OpenProject platform. Microsoft Teams, Viber, and WhatsApp were used for communication within the teams. All the necessary teaching materials and instructions were published on Moodle. All parts of the project were documented and put on OpenProject (Mihajlović-Milićević et al., 2020).

The research was performed in the spring semester of 2019/20 during the coronavirus pandemic. Data collection for the analysis of the applied model was gathered using a questionnaire and the OpenProject tool was used to collect data for SNA analysis.

4. EVALUATION - RESULTS AND DISCUSSION

Managing virtual teams in the development of smart environments, ie. development and implementation of IoT projects were realized using the proposed methodology. 116 undergraduate students and 5 professors participated in the project. Students were asked to organize into virtual teams and complete the project in 4 weeks. The project was complex and considered changes in the organizational and technical environment.

At the end of the project, students and professors answered questions from the questionnaire about their participation in the development of smart environments within virtual teams. A questionnaire containing 26 questions, which measures the estimated performance of the process and the results of the participants' feedback, was completed by 62 students and 2 professors. The Likert scale was used for most questions (grades from 1 - I do not agree to 5 - I completely agree).

Table 1: Survey questions with an average grade

No.	Survey sheet-questions	Average rating
1	The application of the agile framework of the SAFe and DevOps approaches had a positive impact on implementation.	3.94
2	The SAFe framework and DevOps approach provide a higher level of empowerment and collaboration.	4.24
3	Project management could have been more efficient.	2.94
4	Parts of the project solution can be improved.	3.66
5	Team members understood each other's roles.	4.40
6	I met all the team members before starting the project.	3.56
7	Team members initiated communication when solving problems with each other without involving the SAFe Scrum Master or Product Owner.	3.56
8	Communication between team members was easy and successful.	4.56
9	Team members were available to communicate with each other.	4.79
10	Good cooperation and communication between team members is a prerequisite for the realization of the project.	4.89
11	The role of SAFe Scrum Master on the project was clear to me.	4.23
12	Safe Scrum Master was responsible for successfully leading the team.	4.35
13	SAFe Scrum Master influenced the project to be successfully realized.	4.13

No.	Survey sheet-questions	Average rating
14	*SAFe Scrum Master has influenced the product to be of better quality.*	3.95
15	Using the tools OpenProject, Moodle, GitHub provides transparency when working on a project.	4.06
16	The tools OpenProject, Moodle, GitHub are understandable and easy to use.	3.94
17	The possibility of constant review of the work of all team members in OpenProject and GitHub enabled better planning and organization of work.	3.74
18	The tools OpenProject, Moodle, GitHub contains all the functions needed to manage a project.	4.00
19	The OpenProject tool is intuitive and easy to use.	3.92
20	The OpenProject tool enables greater control and easier supervision over the work of all participants in the project.	3.98
21	Using the OpenProject tool has enabled increased productivity.	3.35
22	The OpenProject tool helped make project work more effective and efficient.	3.45
23	The OpenProject tool provided everything we needed.	3.45
24	Using the OpenProject tool is easy to learn.	4.23
25	The OpenProject tool can be used without instructions.	3.73
26	Using the OpenProject tool requires a minimum number of steps to do what is required.	3.85

The results of the study showed that students positively perceived SAFe agile framework and DevOps with average grades of 3.94 and 4.24 (questions 1 and 2, respectively). The role of the SAFe Scrum Master is well received and highly rated (questions 11 to 14). On the control question, ie. the conclusion that the management of the project could have been better (question number 2) was rated low by the students (average score 2.94). Respondents recognized the importance of good communication and assessed that communication on the project was very good (questions 5 to 10). Students rated the tools from *Educational DevOps IoT Lab*: Open Project, Moodle, and GitHub as very useful in project-oriented teaching courses (questions 15 to 18). They also rated OpenProject as a very good grade as a tool for monitoring project activities (questions 19 to 26).

The results of the conducted study also showed that virtual meetings were held in most cases 2-3 times a week and lasted 30 minutes on average. Through communication and cooperation of team members, the processes of exchange and creation of knowledge were identified. The OpenProject tool modules that enable knowledge sharing and creation are News and Wiki. Using the SNA tool (Gephi), the results presented in Figure 3 were obtained.

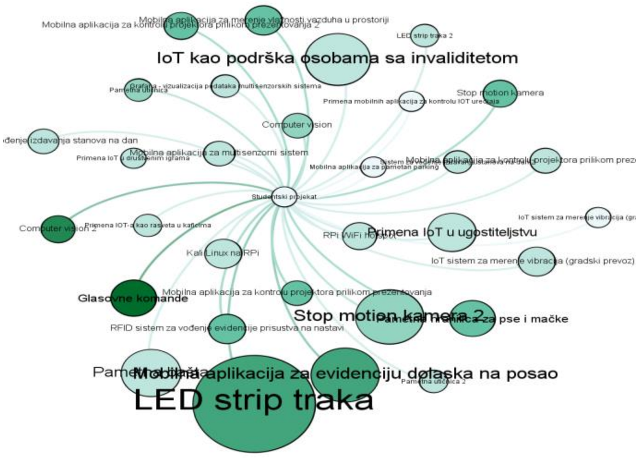
Mobilna aplikacija za kontrolu projektora prilikom prezentovanja 2
Mobilna aplikacija za merenje vlažnosti vazduha u prostoriji
LED strip traka 2
IoT kao podrška osobama sa invaliditetom
Pametna utičnica
Grafana - vizualizacija podataka multisenzorskih sistema
Stop motion kamera
Primena mobilnih aplikacija za kontrolu IOT uređaja
Computer vision
vođenje izdavanja stanova na dan
Mobilna aplikacija za multisenzorni sistem
Primena IoT u društvenim igrama
Mobilna aplikacija za pametan parking
Mobilna aplikacija za kontrolu projektora prilikom prez
Studentski projekat
IoT sistem za merenje vibracija (gr
Computer vision 2
Primena IOT-a kao rasveta u kafićima
RPi WiFi hotspot
Primena IoT u ugostiteljstvu
Kali Linux na RPi
IoT sistem za merenje vibracija (gradski prevoz)
Glasovne komande
Mobilna aplikacija za kontrolu projektora prilikom prezentovanja
Stop motion kamera 2
Pametna hranilica za pse i mačke
RFID sistem za vođenje evidencije prisustva na nastavi
Pametna utičnica 2
Mobilna aplikacija za evidenciju dolaska na posao
LED strip traka

Figure 3: Overview of the use of the Wiki module in the Gephi tool

The Figure 3 shows how much the members of the virtual team used the module and how to display the material within the Wiki module through different shades of green: the strongest shade of green shows the teams that used Wiki the most, up to the lightest shade of green, which means teams that used poorly this possibility. Different sizes of graph nodes, ie teams, represent the extent to which teams posted different contents (video, images, works by other authors ...) within other tool modules that were useful during the project development.

Around 75% of students reported some form of problem regarding the understanding of the whole process and the communication. There were problems in understanding the DevOps process or the role of the SAFe Scrum Master. Scrum masters faced various challenges of coordinating and organizing the work in virtual teams. Students appreciated the opportunity to learn current innovative tools, which could be useful in real work environments, while a smaller group of students thought that the approach was complicated. Most of the students thought that working on such a project was a useful experience and a good way to get acquainted with virtual teams, IoT development, DevOps, and agile methodologies.

From the professor's point of view, the team members managed to master the tools for collaboration and documentation monitoring quite successfully, and the difficulties they encountered since they had to work together in the "virtual world" directed them to each other in a useful way. They managed to establish closer personal and professional ties because mutual trust was built and collective abilities improved.

5. CONCLUSION

The goal of this paper is to propose an effective method for augmenting IT education with DevOps concepts. For this purpose, a conceptual framework was developed, and on top of it, a pipeline made up of different technologies was formulated. The implementation of these pipelines was made to be following academic requirements while trying to maintain the use of best practices and technologies used in the enterprise environments. With this in mind, the proposed model is based on six aspects of DevOps which are: Continuous development, Continuous Integration, Continuous Testing, Packaging, and staging, Release automation, and Performance/relevance monitoring. The end goal of this method is the enrichment of university courses with technologies and knowledge utilized in the industry while providing an education-friendly environment for continual learning and experimentation.

The proposed method was evaluated in practice at the Department of Electronic Business of the Faculty of Organizational Sciences. The evaluation involved two separate groups of students from different subjects: Internet of things and Risk Management in e-business. The utilization of the SAFe framework along with the proposed method of DevOps IoT Lab allowed the students to fill two separate roles, that of project manager, and that of IoT developer, both of these roles had to contend with DevOps concepts, technologies, and pipelines. The participants had the chance to learn various technologies and platforms such as Jenkins, Travis, Tekton, GitHub, Kubernetes, cloud infrastructure services, and IoT Platforms. A strong focus was put on the integration of node-RED with GitHub, and containerization of Node-RED apps with Docker.

At the end of their courses, students and professors were offered a questionnaire about their participation and experiences with the platform. The questionnaire of 26 questions was completed by 62 students and professors, and their reactions towards the platform were generally favorable. The students felt that the platform was technologically empowering and allowed them easier collaboration on their projects. Most students were pleased with the opportunity to learn current innovative tools, which could be useful in enterprise environments, while a smaller group of students felt that this type of work was complicated and unnecessary. In general, students felt that working on such projects was a useful experience and a good way to get acquainted with DevOps and agile development methodologies.

Future research will involve a more thorough evaluation which involves more students, larger projects, and integration of more subjects, such as mobile application development and web development. The different nature of such projects and collaborations between these project groups is bound to challenge the proposed framework and offer new insights. Likewise, new and innovative technologies will be integrated into the pipeline, as well as potential partnerships with the industry in the form of proprietary tools and DevOps expertise.

REFERENCES

Atzori, L., Iera, A., & Morabito, G. (2017). Understanding the Internet of Things: definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, *56*, 122–140.

Bass, L., Weber, I., & Zhu, L. (2015). *DevOps: A software architect's perspective*. Addison-Wesley Professional.

Bayser, M. de, Azevedo, L. G., & Cerqueira, R. (2015). ResearchOps: The case for DevOps in scientific applications. *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 1398–1404. https://doi.org/10.1109/INM.2015.7140503

Bobrov, E., Bucchiarone, A., Capozucca, A., Guelfi, N., Mazzara, M., Naumchev, A., & Safina, L. (2020). *DevOps and Its Philosophy: Education Matters! BT - Microservices: Science and Engineering* (A. Bucchiarone, N. Dragoni, S. Dustdar, P. Lago, M. Mazzara, V. Rivera, & A. Sadovykh (eds.); pp. 349–361). Springer International Publishing. https://doi.org/10.1007/978-3-030-31646-4_14

Bruel, J.-M., & Jiménez, M. (2019). *DevOps'18 Education Panel BT - Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment* (J.-M. Bruel, M. Mazzara, & B. Meyer (eds.), 221–226). Springer International Publishing.

Chatley, R., & Procaccini, I. (2020). *Threading DevOps Practices Through a University Software Engineering Programme*, 90–94.

Kim, G., Debois, P. J., & Willis, J. H. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press. http://library1.nida.ac.th/termpaper6/sd/2554/19755.pdf

Hills, M. (2020). *Introducing DevOps Techniques in a Software Construction Class*. 85–89.

Kuusinen, K., & Albertsen, S. (2019). Industry-Academy Collaboration in Teaching DevOps and Continuous Delivery to Software Engineering Students: Towards Improved Industrial Relevance in Higher Education. *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, 23–27. https://doi.org/10.1109/ICSE-SEET.2019.00011

Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2019). A survey of DevOps concepts and challenges. *ACM Computing Surveys*, *52*(6). https://doi.org/10.1145/3359981

Lopez-Pena, M. A., Diaz, J., Perez, J. E., & Humanes, H. (2020). DevOps for IoT Systems: Fast and Continuous Monitoring Feedback of System Availability. *IEEE Internet of Things Journal*, *7*(10), 10695–10707. https://doi.org/10.1109/JIOT.2020.3012763

Mihajlović-Milićević, J., Naumović, T., & Mitrović, S. (2020). Project-based e-learning in virtual teams. *Proceedings of XVII International Symposium Symorg 2020*, 528–534.

Mihajlović Milićević, J., Filipović, F., Jezdović, I., Naumović, T., & Radenković, M. (2019). Scrum Agile Framework in E-business Project Management: An Approach to Teaching Scrum. *European Project Management Journal*, *9*(1), 52–60. https://doi.org/10.18485/epmj.2019.9.1.7

Ohtsuki, M., & Kakeshita, T. (2019). Utilizing software engineering education support system ALECSS at an actual software development experiment: A case study. *CSEDU 2019 - Proceedings of the 11th International Conference on Computer Supported Education*, *2*(Csedu), 367–375. https://doi.org/10.5220/0007723203670375

Ohtsuki, M., Ohta, K., & Kakeshita, T. (2016). Software Engineer Education Support System ALECSS Utilizing DevOps Tools. *Proceedings of the 18th International Conference on Information Integration and Web-Based Applications and Services*, 209–213. https://doi.org/10.1145/3011141.3011200

Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L. E., Tiihonen, J., & Männistö, T. (2016). *DevOps Adoption Benefits and Challenges in Practice: A Case Study* (P. Abrahamsson, A. Jedlitschka, A. Nguyen Duc, M. Felderer, S. Amasaki, & T. Mikkonen (eds.), 590–597). Springer International Publishing.

Royle, K., & Nikolic, J. (2016). *What we can learn from Agile Work Practices about Learning and Teaching in Schools.* https://doi.org/10.13140/RG.2.1.3501.0161

Yang, D., Wang, D., Yang, D., Dong, Q., Wang, Y., Zhou, H., & Daocheng, H. (2020). DevOps in practice for education management information system at ECNU. *Procedia Computer Science*, *176*, 1382–1391. https://doi.org/10.1016/j.procs.2020.09.148