

PAPER

Mobile DevOps in Education: Practical Training through Application Development

Mykhailo

Luchkevych¹  (✉),
Viktoriia Luchkevych² ,
Iryna Shakleina¹ ¹Lviv Polytechnic National
University, Lviv, Ukraine²Ivan Franko National
University of Lviv,
Lviv, Ukraine[mykhailo.m.
luchkevych@lpnu.ua](mailto:mykhailo.m.luchkevych@lpnu.ua)**ABSTRACT**

This paper investigates the effectiveness of teaching DevOps concepts to students through practical mobile application development. We pay particular attention to the challenges specific to the mobile environment. These include the complexity of deploying applications in stores such as Google Play and the App Store and the need to ensure constant and convenient interaction with mobile users. The paper presents a pedagogical model combining project-based learning, containerization, automated CI/CD processes, and cloud platforms. We adapted these elements to the specifics of mobile development. The research presents examples of student projects that implement DevOps practices. In particular, it describes the automation of testing on different types of mobile devices, the setup of build and deployment processes, and the monitoring of application performance in a cloud environment. This study conducts a comparative analysis between the traditional approach to training and the DevOps-oriented methodology. The indicators considered include update cycle duration, number of errors, and adaptation to mobile platforms. The results confirmed that using DevOps in mobile development education improves the quality of students' technical training, promotes the development of their practical skills, and enhances their competitiveness in the field of mobile technologies.

KEYWORDS

DevOps, mobile development, CI/CD, containerization, cloud computing

1 INTRODUCTION

In today's digital transformation landscape, DevOps is essential for enhancing the efficiency of software development processes. Its implementation promotes automation, provides flexibility, and enhances the reliability of software product creation [1]. DevOps enables teams to respond swiftly to changes, reduce development time, and maintain consistent program code quality.

In light of the increasing demand for DevOps specialists, integrating relevant approaches into higher technical education is vital in training the next generation of IT professionals [2].

Luchkevych, M., Luchkevych, V., Shakleina, I. (2025). Mobile DevOps in Education: Practical Training through Application Development. *International Journal of Interactive Mobile Technologies (IJIM)*, 19(15), pp. 129–141. <https://doi.org/10.3991/ijim.v19i15.55531>

Article submitted 2025-04-04. Revision uploaded 2025-06-04. Final acceptance 2025-06-04.

© 2025 by the authors of this article. Published under CC-BY.

This approach to learning is practical because it aligns with the constructivist paradigm [3]. According to this paradigm, knowledge is formed through active student participation in solving real-world problems [4]. As an educational model, DevOps creates conditions in which students are not only exposed to theoretical material but are also directly involved in designing, implementing, and maintaining software solutions.

This approach aligns with the principles of experiential learning, which is based on a cyclical model of experience, reflection, abstract thinking, and active experimentation [5]. Each project stage ensures knowledge is acquired through practical activities, contributing to a deeper understanding of the material.

DevOps is particularly relevant in mobile technologies, where the development process presents several unique challenges [6, 7]. These include device fragmentation involving various operating systems, screen sizes, and hardware configurations. Deploying mobile applications through app stores creates additional complexity, requiring compliance with clear quality, security, and compliance requirements [8].

From a sociocultural theoretical standpoint [9], DevOps's importance in mobile learning is due to the collective nature of the work. Learning occurs through team interaction, knowledge sharing, and the joint resolution of technical and organizational tasks. This format facilitates students' gradual transition from immediate development to independent professional activity.

Another important aspect is ensuring sustainable and convenient interaction with mobile users. Students studying mobile development should learn to program interfaces and how to automate application testing on different devices [10]. They should also know how to set up CI/CD pipelines [11] to ensure the continuous delivery of updates, use cloud infrastructure to scale services, and analyze user behavioral patterns to optimize application functionality further.

DevOps training based on hands-on mobile development meets the IT industry's requirements and has a theoretical foundation in digital education. The SAMR model [12] can be used to view it, as it distinguishes between four levels of digital technologies in the educational process: 1) substitution, 2) augmentation, 3) modification, and 4) redefinition. The DevOps approach involves more than simply replacing traditional methods with digital tools. Students are transforming the very structure of educational activities. Learning reaches the modification and redefinition levels through automated processes, cloud services, and team collaboration, as defined by the SAMR model.

The structure of DevOps-oriented learning is also consistent with Kohl's theory of experiential learning [13]. This theory includes four key stages: concrete experience, reflective observation, abstract conceptualization, and active experimentation. Implementing student projects in the field of mobile development follows this sequence. Students gain practical experience, analyze the results, form generalizations, and test new approaches in subsequent iterations.

The research in [14] demonstrates a valuable example of the practical application of the constructivist approach in digital education. This study examines the impact of game creation on students' learning of basic programming concepts. The author demonstrated that involving students in creating educational products, such as mobile games in the App Inventor environment, can significantly increase their motivation and interest in learning and improve their learning outcomes. This approach aligns with the principles of experiential learning, in which active student participation in building their own digital experience plays a pivotal role.

A study [15] analyzed the effectiveness of mobile applications in developing computational thinking and programming skills in preschool and primary

school children. The author emphasizes that computational thinking and coding are fundamental 21st-century skills. Teachers should develop these skills in students from an early age. Mobile applications such as ScratchJr and Kodable [16] teach students basic programming concepts. Interactive games and methods help achieve this by building students' logical thinking, creativity, and problem-solving skills.

The proposed teaching approaches are directly related to DevOps principles. In particular, they focus on integration, automation, and continuous improvement. Encouraging students to create their digital products from an early age establishes the cognitive basis for them to master DevOps practices further. The methodology includes collaborative software development, testing, and deployment. Thus, the methods described in [14–16] can serve as the first stage in developing future professionals who are capable of working effectively in a DevOps environment.

Combining hands-on mobile application development with DevOps approaches is an effective way to develop the digital competencies required by modern professionals [17]. This approach covers the entire lifecycle of a mobile product. The DevOps process includes development, testing, containerization, automated deployment, performance monitoring, and collecting user feedback. The latter enables rapid improvement of the application.

Training in this format helps to develop both technical knowledge and soft skills. These include teamwork, critical thinking, and adaptability to rapidly changing IT conditions.

This paper analyzes DevOps learning methodologies through the lens of mobile development. We pay particular attention to the specifics of the mobile environment, including the challenges of deploying applications to app stores and the importance of creating a positive user experience.

It describes the main tools used in the educational process and provides examples of real student projects implementing DevOps practices. The results demonstrate that a mobile-oriented DevOps education effectively trains competitive IT professionals. Such graduates can solve technical and user-related problems in modern mobile development.

2 MATERIALS AND METHODS

The training course, integrated into the bachelor's degree program in Distributed Information Technology, lasted two academic semesters (30 weeks). While most students had basic programming experience and a general understanding of mobile development (e.g., Android Studio) at the beginning of the course, only 18% had previous experience with DevOps practices. We evaluated projects according to clearly defined criteria, including the quality of mobile application implementation, level of process automation (CI/CD), use of cloud infrastructure, monitoring implementation, and effectiveness of team cooperation. We based the final grade on a technical audit of the project (40%), a presentation (30%), and a reflective student report (30%). Students were given compulsory DevOps tasks, such as setting up a CI/CD pipeline, containerizing the backend, and implementing performance monitoring. At the same time, they were given autonomy in choosing technologies, application architecture, and development priorities, helping them to develop independence and creativity.

DevOps training in the context of mobile development is based on a set of interrelated methods. These methods contribute to developing technical skills and

solving specific problems in the mobile environment. Such problems include device fragmentation, the complexity of deploying to official app stores, and the need to maintain a consistent end-user experience.

One of the main approaches is project-based learning [18]. The project-based learning approach involves developing mobile applications based on real-life or training scenarios in teams. Students experience the entire DevOps process, from analyzing requirements and selecting technologies to implementing a mobile product using CI/CD, automated testing, and deployment [19].

During the training, particular attention is given to adapting to different mobile platforms (Android and iOS), considering device fragmentation, and creating user-friendly interfaces. Students also gain experience in customizing the deployment of applications through app stores such as Google Play and the App Store [20]. This approach enables us to model conditions as close as possible to real industrial projects.

Another important training program component is containerization and orchestration [21]. These practices ensure the implementation of microservice architecture in mobile infrastructure. Students learn how to containerize the server side of applications using Docker. They create separate containers for REST APIs, databases, authentication services, and other functional components.

Using Kubernetes enables students to become familiar with the principles of automated scaling, load balancing, and disaster recovery [22]. These skills are particularly relevant for mobile applications with high traffic volumes and stringent reliability requirements.

Automating CI/CD processes is an important part of DevOps training [23]. In the context of mobile development, this involves setting up a mobile application build, running automated tests, and checking compatibility with different operating system versions. It also involves transferring the build to cloud environments for further deployment or direct upload to app stores.

The monitoring tools used include Prometheus, Grafana [24], and the ELK stack [25]. Prometheus collects and stores metrics about the state of the system. Grafana is used to visualize these metrics. It enables the creation of dashboards to monitor the key parameters of mobile applications. The ELK stack—comprising Elasticsearch, Logstash, and Kibana—provides centralized logging and analysis of event logs. The monitoring system allows students to work with large volumes of logs, identify the causes of errors and crashes, and improve software stability.

Tools such as GitHub Actions, GitLab CI/CD, and Jenkins enable students to implement comprehensive CI/CD pipelines [26]. Within these pipelines, they can automate user interface testing using the Espresso or XCTest frameworks. Students also test the performance of mobile applications on both emulators and real devices.

Cloud technologies such as AWS, Azure, and Google Cloud provide the infrastructure for deploying and scaling mobile backends [27, 28]. These technologies also serve as platforms for storing data, user files, and logs. Students are introduced to ‘infrastructure as code’ as part of the educational process. DevOps practitioners use tools such as Terraform or CloudFormation for this purpose. Students learn to deploy APIs, authentication services, and analytical modules like Firebase Analytics that interact directly with mobile clients.

The curriculum implements all of these approaches as part of a comprehensive program. This model of typical industrial practices closely resembles the real production environment. This approach enables students to master DevOps tools and create complete mobile products. These products go through all life cycle stages, from development and placement in the app store to support in the production environment.

The practical experience helps build professional confidence and prepares students to work on complex IT projects.

Figure 1 illustrates the relationship between the key elements of the DevOps learning process adapted for the mobile environment. The model incorporates DevOps practices, including continuous integration/continuous delivery (CI/CD), containerization, and cloud services. At the same time, it addresses the specific challenges of mobile development. These challenges include device fragmentation, application deployment through the App Store or Google Play, and constant interaction with mobile users.

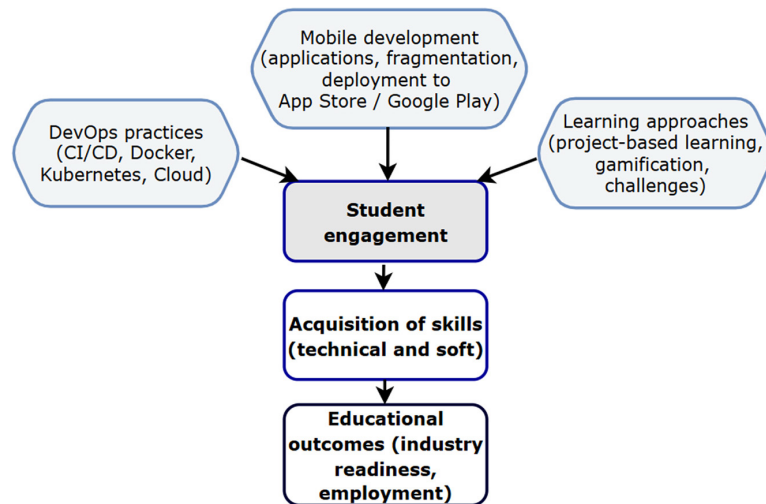


Fig. 1. A conceptual model of DevOps training through mobile development

Educators implement these technologies using appropriate pedagogical approaches. These include project-based learning, incorporating gaming elements, and competitive educational tasks [29]. Using such methods increases student engagement in the educational process.

Increased engagement positively impacts the development of technical and soft skills. In particular, students develop the ability to work in a team, manage DevOps pipelines, deploy mobile applications effectively, and interact with cloud infrastructure.

Consequently, this learning model contributes to achieving educational goals. Students demonstrate their readiness to perform tasks in real-world production environments, increasing their competitiveness in the mobile technology labor market.

3 RESULTS

3.1 Practical cases of DevOps integration into student projects

One of the most effective approaches to teaching modern software development and deployment technologies is the practical implementation of DevOps in student projects [30]. Integrating DevOps into courses enables students to grasp the theoretical principles and acquire hands-on experience in process automation, containerization, monitoring, and cloud platforms.

Several examples of DevOps use were implemented as part of the educational process to demonstrate the effectiveness of this approach.

One example is the automation of a mobile application release using CI/CD tools. Students developed a mobile application to support the learning process as part of the course. The application provided functionality for viewing class schedules, receiving notifications of changes, and facilitating interaction between students and teachers.

We implemented an automated CI/CD pipeline to optimize the development process and accelerate the release of updates. Students used Jenkins and GitHub Actions to set up continuous integration. This automated system offered code reviews, test runs, and deployment of updates, eliminating the need for manual intervention. This approach reduced the time to release new versions of the application by 40%, improved its quality, and reduced the risk of errors in the production environment.

A 40% reduction in update cycle time suggests increased operational efficiency among teams comprising technical students. From an educational perspective, this suggests that the participants in the experiment not only mastered DevOps tools but also acquired the skills to organize teamwork effectively. They also acquired the skills to organize teamwork effectively, reducing the time from idea generation to implementation. This result indicates students' increased readiness for professional activities in the modern IT environment. In such conditions, adapting quickly to changes and implementing technical solutions is crucial.

Additionally, the observed decrease in update cycle duration is directly related to the relevance of students' knowledge and skills to industry needs. It demonstrates future professionals' ability to function effectively in a dynamic software development environment. This ability meets modern employers' expectations of young professionals. It also contributes to a better alignment between academic training and the industry's practical requirements, positively impacting graduates' competitiveness in the labor market.

Another example involves containerizing the server side of a mobile application. The student project involved creating a backend using a microservice architecture. Docker and Kubernetes were then used to build a flexible, scalable, and fault-tolerant environment.

Containerization accelerated the deployment of new server-side versions, cut infrastructure maintenance costs, and enhanced security. Kubernetes ensured automatic load balancing and monitoring of service status. This approach ensured the stable operation of the mobile application, even as the number of active users increased.

Monitoring the performance of mobile applications in a cloud environment is an important aspect of DevOps implementation in student projects. As part of their work, the students implemented a system to analyze the performance of a mobile application. To this end, they employed the Prometheus and Grafana tools.

These solutions enabled them to collect and visualize application performance indicators as analytical dashboards. We used this approach to identify bottlenecks in the software and respond to potential problems.

Implementing the monitoring system provided students with a practical understanding of logging principles, metric collection, and automatic alert setup. This experience helped them become more proficient in using the tools required to maintain mobile applications' stability and smooth operation.

Consequently, we enhanced the software product's reliability and shortened the response time to errors occurring during operation.

These examples demonstrate the effectiveness of integrating DevOps into the educational process. Working with real-life cases and applying modern technologies and automation tools creates conditions that closely resemble professional software development.

Mastering DevOps approaches helps students to develop relevant technical competencies. As a result, the program provides graduates with high-level training and boosts their competitiveness in the modern IT labor market.

3.2 Analyzing the effectiveness of the approach

An analysis of the results of implementing DevOps-oriented training showed its positive impact on students' technical and professional development. According to the final questionnaire results and project defense, 85% of students successfully incorporated DevOps components (CI/CD, containerization, and cloud services) into their learning tasks.

Table 1 summarizes the analysis results, reflecting the quantitative and qualitative changes to the learning process and student project implementation following the integration of DevOps components.

Table 1. Results of DevOps implementation in student projects

Analysis Parameter	Before DevOps Implementation	After DevOps Implementation	Changes (%)
Time to release updates	10 days	6 days	-40%
Average number of code errors per release	15	7	-53%
Infrastructure setup time	5 days	2 days	-60%
Student satisfaction level	70%	92%	+22%

Thus, teaching DevOps through hands-on mobile development has proven effective in technical and professional terms. Students acquire fundamental theoretical knowledge and gain practical experience, facilitating their rapid adaptation to the requirements of the modern IT market.

A correlation analysis was conducted to quantify the relationship between the individual components of DevOps education and student performance. The results indicate a moderate positive correlation between the intensity of CI/CD practice use and project final grades ($r = 0.58$; $p < 0.01$). A significant relationship exists between actively using cloud services and team effectiveness in project work ($r = 0.63$; $p < 0.01$), indicating the integrative impact of DevOps competencies on interdisciplinary learning outcomes.

A comparative analysis with control groups (not using the DevOps approach) determined the effect size (Cohen's d) to be 0.84 to improve the average grade of student projects. According to Cohen's classification, this indicates a significant effect, confirming the tangible advantage of the implemented approach over traditional teaching methods.

The statistical significance of the results was assessed using an independent t-test, which compares the mean values of two independent samples: the main group and the control group. The main group consisted of 52 students who studied mobile development using DevOps approaches. The control group ($n = 47$) followed traditional educational methods without DevOps integration. The average final project scores on a 100-point scale were 88.4 and 81.1 for the primary and control groups, respectively. The t-test revealed a statistically significant difference between these scores ($t(97) = 3.72$; $p < 0.001$), indicating the reliability of the results. Therefore, introducing a DevOps-oriented approach to teaching significantly improves students' academic performance by enhancing the quality of project tasks in mobile development.

A diagram was created to visually represent the quantitative results of the analysis of the effectiveness of DevOps implementation in mobile development education (Figure 2). This table illustrates the key statistical indicators that allow us to evaluate the relationship between various elements of the educational process and student performance, as well as the overall impact of the implemented approach.

Thus, using the analytical apparatus enables positive educational outcomes to be recorded and confirms the presence of statistically significant effects associated with the introduction of DevOps in mobile development. These data provide a reasonable basis for scaling the model further to other educational contexts.

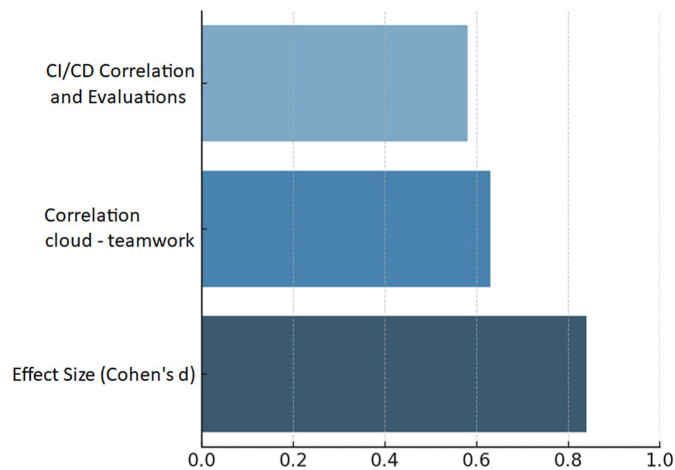


Fig. 2. Key statistical indicators of the effectiveness of DevOps implementation in mobile development education

4 DISCUSSION

Integrating DevOps into learning mobile development demonstrates significant advantages over traditional educational approaches. The main reasons for this are the high level of student engagement, the course's practical nature, and the technology's relevance. Combining DevOps with mobile technologies enables students to master the tools and better understand the real challenges modern mobile application developers face.

One of the key features of the mobile environment is the high level of device fragmentation. This variety includes different screen sizes, operating systems, and hardware configurations. Students must consider these factors in case studies when setting up automated testing and organizing deployment processes. This experience helps students develop the practical skills needed to adapt applications to different mobile devices and environments.

Another important aspect is the complexity of deploying mobile applications in app stores. This process requires compliance with technical, security, and legal requirements. Throughout the program, students learn how to create CI/CD pipelines that automate the build, testing, and deployment of mobile software. These pipelines consider the particularities of the Google Play Console and App Store Connect.

This experience has important practical value. Students gain an understanding of the entire process involved in releasing a mobile app, from making code changes to publishing an update in the app store and its subsequent appearance on the end user's device.

In addition to developing technical skills, DevOps-oriented training helps students understand agile development management methodologies. These include Agile, Scrum, and Kanban, which are implemented in mobile teams with short iteration cycles.

Constant interaction between the development, testing, and support teams recreates the conditions of a real DevOps environment. Such an environment requires precise coordination, transparent processes, and effective communication.

However, implementing DevOps in mobile development training presents particular challenges. One such challenge is the need for technical resources. Students must have access to powerful computers, cloud emulators, and signed developer accounts with Google or Apple.

Additionally, the complexity of the infrastructure, particularly when working with Kubernetes, Docker, and CI/CD systems, can pose a challenge to students without administrative experience.

To overcome these barriers, training materials must be adapted to suit students' levels. Ongoing support from teachers or mentors is also advisable.

In summary, DevOps training in mobile development is a promising educational approach. It ensures the acquisition of technical knowledge and the development of the ability to solve practical problems related to the life cycle of mobile applications.

This training format produces specialists who can create software products and effectively manage their development, implementation, and maintenance in the dynamic environment of digital platforms.

The proposed approach to teaching DevOps through hands-on mobile development is flexible and adaptable to different educational contexts. Educators can implement it in various settings, including universities, colleges, vocational schools, corporate training programs, and online courses. The basic principles—project-based learning, integration of cloud services, CI/CD, and containerization—can be scaled to suit students' training levels. For instance, professional training or programming boot camps could emphasize release automation, mobile optimization, and user data analytics. In contrast, a school or distance learning environment could use simplified development environments and ready-made templates for mobile applications. This adaptability enables the model to form the basis of DevOps-oriented learning paths in various formats and environments.

The DevOps learning model for mobile development can be easily scaled and adapted to different educational formats. In the context of online courses, for example, it can be implemented using cloud platforms to enable remote access to infrastructure and video tutorials on how to set up CI/CD pipelines and virtual laboratories. In a hybrid learning format combining face-to-face and remote interaction, asynchronous DevOps tasks, team projects supported by cloud repositories, and regular online sprints to present progress are advisable. Additionally, this model can be adapted for industry certification programs, emphasizing the practical application of Jenkins, Docker, Kubernetes, and GitLab CI/CD tools, with mandatory demonstrations of mobile application-based cases. Thanks to its flexible structure, this educational model can meet the needs of academic and corporate environments, ensuring the acquisition of relevant competencies in a real-world technological context.

Educators can effectively implement the DevOps approach in various learning formats. In flipped classrooms, for example, teachers can provide students with videos or interactive mini-courses on topics such as setting up CI/CD, Docker, or mobile deployments for self-study. During face-to-face meetings, they can focus on discussing results, solving DevOps problems, and reflecting on the solutions applied. In an interdisciplinary setting, computer science teachers can collaborate with colleagues

from project management or design courses to organize complex team projects. One group of students might develop a mobile front end, another might develop a DevOps infrastructure for its deployment, and a third group might conduct usability testing and performance analysis. This approach simulates real-world production conditions and promotes teamwork, critical thinking, and an understanding of the role of DevOps in the entire life cycle of digital products.

5 CONCLUSION

Integrating DevOps into the mobile application development curriculum has proven an effective way of preparing students for the real challenges of the modern IT industry. Practice-oriented learning contributes to both the acquisition of theoretical knowledge and the development of skills in working with the main DevOps tools. These include CI/CD pipelines, containerization tools, cloud services, and software application performance monitoring tools.

A key advantage of this approach is that it creates conditions that closely resemble real production processes. This approach provides students with a comprehensive understanding of the software life cycle. They learn to apply their knowledge confidently in practical tasks.

Working with CI/CD pipelines helps students understand the importance of automated testing and deployment processes. Working with containerization and orchestration technologies enables students to grasp the principles of scalability, fault tolerance, and the efficient management of resources in modern applications.

In addition to technical competencies, DevOps training in a mobile context contributes to developing soft skills. These include flexible thinking, teamwork, adaptability to a dynamic environment, and readiness for change.

Graduates who have mastered this educational model are better prepared for their future careers. Employers in Ukraine and abroad highly value their knowledge and practical experience.

Despite its many advantages, further research should focus on optimizing the educational process. One promising area is expanding the use of DevOps in distance learning. Combining mobile technologies with DevOps approaches in a remote learning environment enables continuous learning, regardless of the student's geographical location.

Another important development is integrating artificial intelligence tools to automate educational processes. This approach includes intelligent monitoring of learning progress, adaptive adjustment of task complexity based on individual skill levels, and automatic optimization of DevOps learning pipelines.

Using such solutions can significantly increase the flexibility and efficiency of educational programs. This approach allows training to be better adapted to the needs of students with varying initial knowledge and practical skills.

In addition to these areas, the following aspects are promising for further research:

- Longitudinal tracking of the career trajectories of graduates who have studied educational programs integrating practical DevOps components into mobile development. This will enable us to evaluate the long-term impact of this methodology on graduates' employment, professional development, and adaptability to the demands of the IT sector;
- Controlled experimental studies will compare the results of traditional education with those of advanced programs incorporating DevOps practices. This will

enable us to quantify the effectiveness of new educational models across various academic settings;

- Analyzing the impact of the choice of mobile platform (Android, iOS, or multi-platform solutions) on the specifics of DevOps implementation in student projects. Particular attention should be paid to problems such as device fragmentation, the complexity of environment configuration, and the optimization of automatic testing and deployment processes;
- Another key aspect of the study is exploring motivational approaches that combine elements of gamifying DevOps tasks and mobile learning. Such models can increase student engagement and promote better learning.

In general, incorporating DevOps into mobile development education is a promising approach. Not only does it improve the quality of technical education, but it also ensures that competitive specialists capable of responding to the current challenges of the digital economy are trained and capable of responding to the current challenges of the digital economy.

In-depth empirical research in this area would significantly expand the scientific understanding of effective educational strategies for training the next generation of IT professionals.

6 ETHICAL APPROVAL STATEMENT

Lviv Polytechnic National University's ethical standards were implemented in all of the educational activities described in this article. Data were collected and analyzed by the principles of academic integrity, respect for the dignity of participants in the educational process, and the ethical treatment of human subjects.

The student projects and related learning activities described in the article have received official approval from the Department of Information Systems and Networks at Lviv Polytechnic National University (Protocol No. 17, 30 April 2025). As the study did not involve using personal or sensitive data nor pose any risk to participants, an exemption from the requirement to obtain written informed consent was officially granted. Nevertheless, all students were informed about their participation in the educational experiment and the purpose of the study, and they were given the opportunity to withdraw from the study at any time without facing any negative consequences.

The department confirms that all research procedures comply with the ethical provisions of the Declaration of Helsinki (World Medical Association) and current international standards governing research involving human subjects in an academic environment.

7 REFERENCES

- [1] R. Amaro, R. Pereira, and M. Mira da Silva, "DevOps metrics and KPIs: A multivocal literature review," *ACM Computing Surveys*, vol. 56, no. 9, pp. 1–41, 2024. <https://doi.org/10.1145/3652508>
- [2] M. Fernandes *et al.*, "Devops education: An interview study of challenges and recommendations," in *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Software Engineering Education and Training*, 2022, pp. 90–101. <https://doi.org/10.1145/3510456.3514152>

- [3] A. MacLeod, S. Burm, and K. Mann, "Constructivism: Learning theories and approaches to research," in *Researching Medical Education*, J. Cleland and S. J. Durning, Eds., 2022, pp. 25–40. <https://doi.org/10.1002/9781119839446.ch3>
- [4] L. M. Angraini, N. Kania, and F. Gürbüz, "Students' proficiency in computational thinking through constructivist learning theory," *International Journal of Mathematics and Mathematics Education*, vol. 2. no. 1, pp. 45–59, 2024. <https://doi.org/10.56855/ijmme.v2i1.963>
- [5] L. Y. Jonathan and M. N. Laik, "Using experiential learning theory to improve teaching and learning in higher education," *European Journal of Education*. vol. 7, no. 2, pp. 18–33, 2024. Retrieved from https://revistia.com/files/articles/ejed_v7_i2_24/Jonathan.pdf
- [6] B. O. Kose, "Mobilizing DevOps: Exploration of DevOps adoption in mobile software development," *Kybernetes*, 2024. <https://doi.org/10.1108/K-04-2024-0989>
- [7] R. Eramo *et al.*, "An empirical study on the role of devops in the development of mobile applications," *SSRN*, 2024. <https://doi.org/10.2139/ssrn.5031770>
- [8] W. Zhu *et al.*, "What is an app store? The software engineering perspective," *Empir. Software Eng*, vol. 29, no. 35, p. 35, 2024. <https://doi.org/10.1007/s10664-023-10362-3>
- [9] G. Chan *et al.*, "A panoramic view of socio-cultural sensitivity in digital technologies: A comprehensive review and future directions," *International Journal of Human–Computer Interaction*, vol. 41, no. 4, pp. 1917–1945. <https://doi.org/10.1080/10447318.2024.2372135>
- [10] N. G. Berihun, C. Dongmo, and J. A. Van der Poll, "The applicability of automated testing frameworks for mobile application testing: A systematic literature review," *Computers*. vol. 12, no. 5. p. 97, 2023. <https://doi.org/10.3390/computers12050097>
- [11] L. Nguyen Ba, "Enterprise-grade CI/CD pipeline for mobile development," 2022. Retrieved from <https://www.theseus.fi/handle/10024/753156>
- [12] M. T. Zamri and S. N. A. Mohamad, "Technology integration in education: A review and analysis of SAMR model," *International Journal of Research and Innovation in Social Science*, vol. VIII, no. IIS, pp. 6195–6200, 2025. <https://doi.org/10.47772/IJRIS.2024.803463S>
- [13] T. H. Morris, "Experiential learning – a systematic review and revision of Kolb's model," *Interactive Learning Environments*, vol. 28, no. 8, pp. 1064–1077, 2019. <https://doi.org/10.1080/10494820.2019.1570279>
- [14] S. Papadakis, "Evaluating a game-development approach to teach introductory programming concepts in secondary education," *International Journal of Technology Enhanced Learning*, vol. 12, no. 2, pp. 127–145, 2020. <https://doi.org/10.1504/IJTEL.2020.106282>
- [15] S. Papadakis, "Apps to promote computational thinking concepts and coding skills in children of preschool and pre-primary school age," in *Mobile Learning Applications in Early Childhood Education*, Stamatios Papadakis and Michail Kalogiannakis, Eds., 2020, Hershey, Pennsylvania, USA: IGI Global Scientific Publishing, pp. 101–121. <https://doi.org/10.4018/978-1-7998-1486-3.ch006>
- [16] S. Papadakis, "Apps to promote computational thinking and coding skills to young age children: A pedagogical challenge for the 21st century learners," *Educational Process: International Journal (EDUPIJ)*, vol. 11, no. 1, pp. 7–13, 2022. <https://doi.org/10.22521/edupij.2022.111.1>
- [17] A. Gupta, "A case study: DevOps transformation in mobile application products: Journey, challenges, and solutions," *J. Artif. Intell. Mach Learn and Data Sci.*, vol. 1, no. 1, pp. 655–660, 2022. Retrieved from <https://urfjournals.org/open-access/a-case-study-devops-transformation-in-mobile-application-products-journey-challenges-and-solutions.pdf>
- [18] W. Liao, "Project-Based learning: Teaching mobile app development in capstone courses," *Journal of Computing Sciences in Colleges*, vol. 38, no. 3, p. 209, 2022. <https://dl.acm.org/doi/abs/10.5555/3580523.3580570>

- [19] M. Radenković, S. Popović, and S. Mitrović, “Project based learning for DevOps project based learning for DevOps: School of Computing experiences,” *E-business Technologies Conference Proceedings*, vol. 2, no. 1, pp. 127–131, 2022. Retrieved from <https://www.ebt.rs/journals/index.php/conf-proc/article/view/133>
- [20] J. Cowls, J. Morley, and L. Floridi, “App store governance: Implications, limitations, and regulatory responses,” *Telecommunications Policy*, vol. 47, no. 1, p. 102460, 2023. <https://doi.org/10.1016/j.telpol.2022.102460>
- [21] J. Jordanov, D. Simeonidis, and P. Petrov, “Containerized microservices for mobile applications deployed on cloud systems,” *International Journal of Interactive Mobile Technologies*, vol. 18, no. 10, pp. 48–58, 2024. <https://doi.org/10.3991/ijim.v18i10.45929>
- [22] T. Golis, P. Dakić, and V. Vranić, “Automatic deployment to kubernetes cluster by applying a new learning tool and learning processes,” *SQAMIA 2023 Software Quality Analysis, Monitoring, Improvement, and Applications*, vol. 1613, p. 0073, 2023. Retrieved from <https://ceur-ws.org/Vol-3588/p16.pdf>
- [23] M. Á. Sánchez-Cifo, P. Bermejo, and E. Navarro, “DevOps: Is there a gap between education and industry?” *Journal of Software: Evolution and Process*, vol. 35, no. 12, p. e2534, 2023. <https://doi.org/10.1002/smr.2534>
- [24] M. Yudha Erian Saputra, Noprianto, S. Noor Arief, V. Nur Wijayaningrum, and Y. W. Syaifudin, “Real-Time Server Monitoring and Notification System with Prometheus, Grafana, and Telegram Integration,” in *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETISIS)*, Manama, Bahrain, 2024, pp. 1808–1813. <https://doi.org/10.1109/ICETISIS61505.2024.10459488>
- [25] Šaler M. “Sistemski nadzor s ELK stack alatima,” Undergraduate thesis, University of Applied Sciences in Information Technology, 2024. Retrieved from <https://zir.nsk.hr/islandora/object/vsite:19>
- [26] S. T. Makani and S. Jangampeta, “The evolution of CICD tools in DevOps from Jenkins to GitHub actions,” *International Journal of Computer Engineering and Technology (IJET)*, vol. 13, no. 2, pp. 1–11, 2022. Available online at <https://iaeme.com/Home/issue/IJCET?Volume=13&Issue=2>
- [27] F. M. Dahunsi, J. Idogun, and A. Olawumi, “Commercial cloud services for a robust mobile application backend data storage,” *Indonesian Journal of Computing, Engineering, and Design (IJoCED)*, vol. 3, no. 1, pp. 31–45, 2021. <https://doi.org/10.35806/ijoced.v3i1.139>
- [28] M. Luchkevych, I. Shakleina, and O. Duda, “The impact of modern cloud technologies on the efficiency of DevOps processes,” *Scientific Journal of TNTU (Tern.)*, vol. 117, no. 1, pp. 112–122, 2025. https://doi.org/10.33108/visnyk_tntu2025.01.112
- [29] M. Luchkevych, “Active learning methods in teaching DevOps,” *Youth and the Market*, vol. 11, no. 231, pp. 100–105, 2024. <https://doi.org/10.24919/2308-4634.2024.314762>
- [30] M. M. Luchkevych, “Challenges and issues in implementing DevOps curriculum subjects into the educational process,” *Scientific Journal of M. P. Drahomanov National Pedagogical University. Series 5: Pedagogical Sciences: Realities and Perspectives*, vol. 102, pp. 43–47, 2024. <https://doi.org/10.31392/UDU-nc.series5.2024.102.09>

8 AUTHORS

Mykhailo Luchkevych is with the Lviv Polytechnic National University, Lviv, Ukraine (E-mail: mykhailo.m.luchkevych@lpnu.ua).

Viktoriia Luchkevych is with the Ivan Franko National University of Lviv, Lviv, Ukraine.

Iryna Shakleina is with the Lviv Polytechnic National University, Lviv, Ukraine.