WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Research on the Necessity of Implementing DevOps Technologies in the Training of Future Computer Science Teachers

Sarthak Srivastava[1]

[1]*Visa Inc, United States*

DOI: https://doi.org/10.55248/gengpi.4.923.92505

A B S T R A C T

In the article, the problem of implementing DevOps technologies in the education of future computer science teachers is explored. This issue has arisen due to the development and expansion of digital technologies and increased demands from stakeholders for future computer science teachers. Through scientific analysis and the systematic categorization of academic publications, the current state of DevOps technologies and their impact on the process of digitization and digitalization of society were examined. It was determined that the professional IT community actively adopts and promotes DevOps technologies. The analysis of publications revealed that there are currently few educational and professional programs that include the study of DevOps. Educational and professional programs in the field of "Secondary Education (Computer Science)" were specifically noted, with most of these programs lacking elements of DevOps. Modern directions for improving the content of school computer science courses focus on enhancing their practical orientation, and DevOps technologies can contribute to this. The study identified several substantive components of DevOps technologies that can be integrated into the preparation of computer science teachers, including infrastructure as code, configuration management, containers, container orchestration, infrastructure security, deployment pipelines, microservices architecture, post-production considerations, and domain-specific DevOps peculiarities. The inclusion of DevOps elements in the training of future computer science teachers should be based on stakeholder needs. Computer science teachers do not need to master all technical and technological aspects of DevOps implementation and application, but they should possess a sufficient level of professional competencies for successful employment. The results of the confirmatory experiment conducted in this study highlighted the necessity of teaching DevOps technologies to future computer science teachers. Stakeholders also identified the most relevant DevOps technologies for modern computer science teachers, including infrastructure as code, containers, and container orchestration..

Keywords: Secondary education, computer science teacher, educational program, DevOps, professional training.

1. Introduction

DevOps Professional training for future computer science teachers must align with contemporary trends in information technology, programming, and network technologies. Achieving this alignment is possible through continuous refinement and updating of the curriculum. One such technology is DevOps. Researchers A. Dyck, R. Penners, and H. Lichter provided the following definition of DevOps in their study (Dyck, Penners, Lichter, 2021): "DevOps is an organizational approach grounded in collaboration and cross-functional cooperation within and between development (programming) and IT operations teams in software-producing organizations to manage stable systems and accelerate change."

According to a survey conducted in 2021 (State of DevOps Report, 2021, p. 6), 78% of software development organizations have introduced elements of DevOps culture into their work, while 18% have fully implemented DevOps. There is currently a high demand for DevOps engineers. In 2022, according to a DevOps Institute survey, a shortage of IT skills was identified as a significant global issue, with 40% of respondents indicating that the scarcity of resources and skills is one of the top three challenges today. Additional research confirms a significant skills gap in the field of technology and IT worldwide (Oehrlich & Settle, 2022, p. 7). DevOps is on its way to becoming a necessary skill for IT practitioners. Therefore, elements of DevOps technologies should be integrated into the professional activities of modern computer science teachers in the near future. The professional IT community provides numerous blogs, webinars, conferences, communities, and organizations that seek to popularize, promote, support, and enhance DevOps. However, research in this field is relatively limited. The integration of DevOps technologies into the curriculum for IT majors already has some experience, but educational and professional programs dedicated to DevOps are relatively rare. If educators have limited knowledge of DevOps, students are unlikely to acquire the necessary DevOps skills for an advanced information society. Therefore, the current state of education for higher education students will not be able to meet the demand for numerous DevOps vacancies. Educational and professional programs for higher education students in the field of "14.09 Secondary Education (Computer Science)" do not provide opportunities for studying DevOps technology, as evidenced by the lack of scientific research on this issue. Thus, there is a contradiction between society's need for future computer science teachers to be knowledgeable about DevOps technologies and the absence of DevOps elements in the curriculum for students majoring in "14.09 Secondary Education (Computer Science)."

The research's aim is to analyze the problem and provide justification for the incorporation of DevOps elements into the curriculum for future computer science teachers.

Research Methodology:

The study employed content analysis of scientific publications related to the identified problem, as well as the synthesis and categorization of existing approaches to teaching DevOps to higher education students. Pedagogical confirmatory experiments were conducted to explore the requirements of stakeholders regarding the acquisition of DevOps skills by future computer science teachers. Descriptive statistical methods were used to analyze the results of the confirmatory experimental research.

Research Findings:

Unlike other topics in the field of information technology, DevOps is much more than just a technology. IT experts D.A. Tamburri and D. Perez-Palacin argue that DevOps represents a cultural, procedural, and technological movement (Tamburri & Perez-Palacin, 2018). Therefore, studying DevOps differs from learning any other technology. To meet the existing demand for introducing future computer science teachers to DevOps technology, it is necessary to develop theoretical foundations for the training of higher education students in the field of DevOps. Currently, there are almost no educational and professional programs that include DevOps disciplines. Despite the significant benefits that DevOps brings to software development (State of DevOps Report, 2021; Oehrlich & Settle, 2022; Forsgren & Humble, 2015), researchers do not use DevOps in their research projects involving software development. This is due to the high cost of implementing DevOps, which is not justified for research purposes. Companies hiring DevOps specialists do not have well-defined competencies for these professionals, even though DevOps technology is highly popular and in great demand. Employers expect DevOps professionals to have competencies in all related fields, which complicates the development of corresponding educational programs. In conclusion, the research emphasizes the necessity of integrating DevOps elements into the education of future computer science teachers, given the cultural, procedural, and technological significance of DevOps. It also highlights the existing gap between the demand for DevOps skills in the job market and the limited availability of DevOps education in current programs. The DevOps community is evolving rapidly, making it challenging to establish a unified approach to training professionals (Vicente & Cunha, 2022). Available DevOps certification programs, courses, conferences, and communities often struggle to reach a consensus on professional preparation in this field. To understand the prerequisites and the current state of DevOps application for automating software development processes, the study aimed to identify elements that could be integrated into the curriculum for future computer science teachers. Initially, software developers often moved towards modular integration closer to the end of a project, leading to frequent breakdowns and significant developer frustration (Duvall, Matyas & Glover, 2007). This situation prompted the adoption of Continuous Integration (CI), a practice that involves integrating modules throughout the project's lifecycle to avoid complex integration challenges and numerous errors in the project's final stages.

Most IT professionals associated the concept of CI with Extreme Programming (Tamburri & Perez-Palacin, 2018). Towards the end of the 1990s, the IT industry, particularly organizations with significant resources, recognized the necessity and value of implementing CI to address the rapidly growing complexity of software development. In the book "Microsoft Secrets" (Cusumano & Selby, 1998), M.A. Cusumano et al. described how Microsoft adopted CI and benefited from it by "developing everything in parallel, with frequent synchronizations." Microsoft referred to the model as synchronized and stabilizing, effectively implementing CI. During that period, IT professionals also used various terms to describe the synchronized model, including "daily build," "nightly build," or "zero defects." The first comprehensive study that fully described the practice and implementation of CI was published by P.M. Duvall, S. Matas, and A. Glover (Duvall, Matyas & Glover, 2007). The evolving landscape of DevOps, the complexity of its components, and its transformative impact on software development highlight the need for a structured approach to introducing DevOps concepts into the curriculum for future computer science teachers. Understanding the historical context and the development of practices such as CI can inform this process and help bridge the gap between industry demands and educational offerings.

2. Research on CI and CD in DevOps

The researchers identified six tasks within CI:

Continuous integration of code.

Continuous integration of databases.

Continuous testing.

Continuous inspection.

Continuous delivery.

Continuous feedback.

These six tasks primarily focused on integration and programming. N. Forsgren and J. Humble, in their work, studied the second part of DevOps technologies, which is Continuous Delivery (CD) (Forsgren & Humble, 20160). The authors described the impact of continuous delivery practices in organizations on existing technological processes. They argued that the use of CD positively affects technical professionals and enhances the software delivery productivity, reducing software failures. The study also noted that CD has an indirect impact on overall organization efficiency through IT productivity. Therefore, DevOps is not just about technologies; it encompasses a culture of software development, shared practices, and automation that align development and IT operations teams to follow a unified approach for improving customer experience, responding quickly to business needs, and balancing innovation with security and information system administration requirements. The DevOps workflow is well-known to IT professionals and includes all tasks and processes of CI and CD. The CD process consists of two parts - deployment and release. In addition to CI, DevOps includes many other continuous processes, such as continuous planning, continuous utilization, continuous trust, continuous improvement, continuous innovation, and more. Technically, DevOps does not cover all continuous software development processes. The goal of DevOps is to unite the roles of developers and administrators to streamline the software delivery process and ensure collaboration throughout the software development lifecycle. This holistic approach improves software quality, accelerates development cycles, and enhances the overall efficiency of the organization.

Usage and Research Trends in DevOps:

DevOps is a part of software engineering that has gained significant popularity in the IT industry. Its popularity has grown among IT professionals over the last decade (State of DevOps Report, 2021) and has also captured the attention of researchers in the field (Tamburri D.A. & Perez-Palacin, 2018; Azad & Hyrynsalmi, 2022; Sánchez-Cifo, Bermejo & Navarro, 2023; Amaro, Pereira & Mira da Silva, 2022). However, the number of academic publications on the subject of DevOps is relatively low, especially within the Ukrainian research segment. The popularity of DevOps in the industry can be observed in three ways: publication trends, the creation of supporting tools, and a significant number of surveys with a large number of participants. Many ommunities, user groups, and DevOps blogs have been created to inform IT professionals about this subject (Lennon, 2023). The State of DevOps Report (2021) suggests that DevOps practices lead to increased productivity for IT companies. This, in turn, improves business outcomes measured by profitability and market share. Therefore, DevOps has a high return on investment, and companies are willing to invest in popularizing DevOps technologies. In Ukraine, examples of such companies are SoftServ and EPAM.The popularity of DevOps technologies worldwide is evident from the large number of conferences scheduled for 2023 (Best Upcoming DevOps Conferences in 2023, 2023). Now, let's examine DevOps tools and their usage practices to determine the potential for their incorporation into the curriculum for future computer science teachers. Given the high demand for DevOps in the industry, many companies and individuals are developing tools to support the Deployment Pipeline (Gall & Pigni). One such company is Digital.ai, which develops enterprise-scale DevOps tools. The company is a leader in the field of continuous delivery and software release automation. Overall, the growing popularity of DevOps, both in terms of adoption and research, indicates its significance in the IT industry. Incorporating DevOps practices and tools into the curriculum for future computer science teachers can help them stay relevant and better prepare their students for careers in the IT field.

3. Classification of DevOps Tools:

Digital.ai has attempted to track the growing number of DevOps tools and has proposed a classification of these tools based on their licensing and functions (The Periodic Table of DevOps Tools…, 20230). Here is an overview of the different categories of DevOps tools:

Source Code Management: Tools for managing source code revisions.

Repository Management: Tools for managing code repositories.

Database Management: Tools for managing databases.

Configuration Management: Tools for managing configuration or preparing it.

Building: Tools for building applications.

Testing: Tools for automated testing.

Continuous Integration (CI): Tools for continuous integration practices.

Deployment: Tools for deployment automation.

Release Management: Tools for managing software releases.

Logging: Tools for logging and monitoring.

Business Intelligence (BI) or Monitoring: Tools that generate statistics and analytics based on business or system data.

Cloud, Infrastructure as a Service (IaaS), or Platform as a Service (PaaS): Tools that provide shared infrastructure for deploying applications.

Containerization: Tools that create isolated user space instances in the operating system.

Collaboration: Tools that facilitate collaboration among all stakeholders in the software development lifecycle.

Security: Tools focused on security aspects.

A wide range of DevOps tools with various functional capabilities serves the broad spectrum of IT processes in DevOps. There is no single DevOps tool that supports all DevOps functions. However, some categories of tools form the core of the DevOps process. Let's take a closer look at these tools.

Source Code Management tools, which manage code revisions, are among the most crucial tools in DevOps. These tools ensure that changes to the source code are tracked, versioned, and can be easily collaborated on by multiple team members. Examples of source code management tools include Git and Mercurial. These tools play a fundamental role in enabling collaboration and version control, which are essential aspects of modern software development practices. Computer science teachers should be aware of and proficient in these tools to effectively teach their students about contemporary software development methodologies.

Travis CI: Travis CI is a distributed web-based CI tool that builds projects hosted on GitHub and executes predefined build commands from a YAML file.

Jenkins: Jenkins is a server-based CI tool written in Java, running in a servlet container. Jenkins supports various types of source code management tools and can execute different types of build commands.

Containerization: Traditional CI tools build an application and then deploy it on the operating system. Containerization tools create multiple isolated user space instances within the operating system, allowing multiple programs to be deployed in the OS to reduce resource requirements. Docker is a popular containerization tool that automates the deployment of software applications within containers.

Kubernetes: Kubernetes is a popular container orchestration tool, open-sourced by Google, that organizes the deployment, scaling, and management of containerized applications across multiple hosts.

While many different DevOps tools serve a wide range of IT processes, they are not directly related to each other. Therefore, a configuration management tool is needed to configure and manage various system resources. Puppet is an open-source configuration management utility and is one of the leading Infrastructure as Code (IaC) solutions. Puppet has several product lines. These tools are fundamental to DevOps practices and are essential for managing source code, automating build processes, containerization, orchestration, and configuration management. Teachers preparing future IT professionals should consider including these tools in their curriculum to ensure that students are well-equipped with the skills required in the DevOps field.

DevOps products, such as Puppet Discovery, Puppet Enterprise, and Puppet Pipelines, automate the modern code deployment process. We have only discussed the fundamental categories of DevOps tools, but there are many more. Each DevOps pipeline uses tools according to its needs. Therefore, there are numerous opportunities for new DevOps tool providers. Compared to its popularity in the industry, DevOps is not particularly popular in academic circles. An analysis of research in major IT-themed catalogs such as IEEE Xplore, ACM, and Google Scholar shows that DevOps only began to be addressed in research starting from 2008. In contrast, Microsoft fully embraced Continuous Integration (CI) in the development process as early as 1998. It took academic circles more than ten years to consider the concepts of CI, Continuous Delivery (CD), and DevOps after they became popular.

After 2011, the number of published articles has been steadily increasing due to the growing relevance of DevOps in the field. DevOps is a very modern term, so publications on this topic will continue to grow each year. IT professionals and researchers in academic circles have varying levels of interest in DevOps, influenced by different research interests related to DevOps. For example, V. Garousi and M. Felderer compare industrial and academic publications on software testing, which is a subtopic of DevOps (Garousi & Felderer, 2017). Industrial and corporate research tends to focus more on real practices, while academic research leans toward theory. V. Garousi and M. Felderer also argue that IT professionals consider academic publications too formal and complex to understand and implement in software development practice. Researchers note that IT professionals do not find academic research practical or useful. As a result, collaboration between real-world practical projects and academic circles is relatively rare. Without collaboration with the industry, it is challenging to conduct academic research on DevOps since DevOps encompasses all software development processes performed by IT professionals, including programmers, system administrators, operational analysts, support analysts, and more. DevOps also covers all infrastructure management for development, testing, deployment, production, scaling, virtualization, and more. It is unlikely that academic circles can replicate or model complex industrial environments for research. Therefore, collaboration between the IT industry and researchers becomes crucial.

Technologies related to DevOps are beginning to appear in educational disciplines to prepare IT engineers. Researchers R. Hobeck, I. Weber, L. Bass, in their work (Hobeck et al., 2021), investigate the content of subjects taught at engineering faculties in universities. As researchers note, the material introduces students to technical concepts such as microservices architecture, deployment pipelines, or infrastructure as code. On the other hand, practical tasks are offered with standard software tools such as Docker, Kubernetes, Jenkins, or Logstash. Researchers identify that subjects cover topics such as infrastructure as code, configuration management, virtual machines, containers, networking, cloud, container management, infrastructure security, deployment pipelines, microservices architecture, service networks, post-production, disaster recovery, secure development, DevOps specifics for specific domains, and distributed system architecture.

DevOps also encompasses all infrastructure management for development, testing, deployment, production, scaling, virtualization, and more. It is unlikely that academic circles can replicate or model complex industrial environments for research. Therefore, collaboration between the IT industry and researchers becomes crucial. To demonstrate that a specific process or tool brings improvements, academic researchers require collaboration and real data from various deployment pipeline participants.

DevOps technologies are starting to appear consistently in educational disciplines for the preparation of IT engineers. Researchers R. Hobeck, I. Weber, L. Bass, in their work (Hobeck et al., 2021), investigate the content of subjects taught at engineering faculties in universities. As researchers note, the material introduces students to technical concepts such as microservices architecture, deployment pipelines, or infrastructure as code. On the other hand, practical tasks are offered with standard software tools such as Docker, Kubernetes, Jenkins, or Logstash. Researchers identify that subjects cover topics such as infrastructure as code, configuration management, virtual machines, containers, networking, cloud, container management, infrastructure security, deployment pipelines, microservices architecture, service networks, post-production, disaster recovery, secure development, DevOps specifics for specific domains, and distributed system architecture. Let's consider the place of DevOps technologies in the school computer science curriculum. In the curriculum for the 10th-11th grades in computer science (hereinafter referred to as the Program), it is stated that it prepares students for participation in Olympiads, competitions, tournaments, scientific-practical conferences, research competitions of various levels, and other intellectual competitions (Educational program for the profile level..., 2011). Setting such a goal requires teachers to be prepared to use cutting-edge technologies in the field of computer science, including DevOps technologies. The practical skills that the Program aims to develop include: '... skills in analyzing known methods of algorithm construction and determining the most optimal ones for solving specific tasks; skills in testing complex algorithms; skills in working with programming environments; programming technique skills' (Educational Program for the Profile Level..., 2011). Programming and software development are not possible without testing, including automated testing, which is part of modern development environments. Therefore, the ability to use modern development environments and their functionality for automated testing is part of DevOps technologies. Let's examine the content of the Program material that contains elements of DevOps technologies. The topic 'Programming Language and Data Structures' has an activity component 'Creates and executes own test suites and prepared ones,' which involves automated testing of developed algorithms, and this is an element of DevOps technologies.

The topic 'Paradigms and Programming Technologies' is a central theme in the study and application of DevOps technologies in the school computer science course. The knowledge component of this topic involves studying software development methodologies. Modern software development methodologies are based on the complete cycle of implementing DevOps technologies. The activity component of expected learning outcomes for the topic 'Paradigms and Programming Technologies' involves students mastering the complete software development cycle, from design to deployment, which directly transforms the theoretical ideas of DevOps technologies into practical implementation at the school computer science course level.

The preparation of students under the educational-professional program 'Secondary Education (Computer Science)' at Berdiansk State Pedagogical University is carried out in accordance with a list of general and professional competencies.

4. Results of the research.

The study of DevOps technology elements by future computer science teachers should be based on the needs of stakeholders. Computer science teachers do not need to possess all technical and technological aspects of implementing and using DevOps technologies, but they should have a necessary level of professional competencies for future successful employment.

According to the conducted analysis, the following topics remain unaddressed in the educational-professional program "Secondary Education (Computer Science)":

Infrastructure as Code

Configuration management

Containers

Container orchestration

Infrastructure security

Deployment pipelines

Microservices architecture

Post-production

Domain-specific DevOps features

During the research, a descriptive experiment was planned, prepared, and conducted. The research aimed to identify DevOps technologies that are worth teaching to future computer science teachers within the framework of the educational-professional program "Secondary Education (Computer Science)".

To organize and conduct the descriptive experiment, the following tasks were defined:

Determine the circle of respondents who will participate in the survey within the descriptive experiment.

Formulate a list of DevOps technologies that are not yet taught to future computer science teachers.

Identify research methods and relevant criteria to be applied in the descriptive experiment.

Develop a questionnaire for surveying respondents participating in the descriptive experiment.

Conduct surveys of respondents and perform a statistical analysis of the obtained results.

Identify a list of recommended DevOps technologies that are worth teaching in the educational-professional program "Secondary Education (Computer Science)" at Berdiansk State Pedagogical University.

In accordance with the first task, the circle of respondents who would participate in the research was determined. For this purpose, stakeholders of the educational program were invited: teachers and heads of general secondary education institutions, teachers and heads of extracurricular educational institutions, representatives of teacher professional development institutions, representatives of regional education departments.

The research is dedicated to DevOps technologies in the education of future computer science teachers, so the respondents should be familiar with these technologies. In this case, the respondents will be able to correctly assess the need for teaching DevOps technologies to future computer science teachers.

We used a manual method of selecting respondents for the experimental group from the general population. For this purpose, an initial survey of stakeholders was conducted to determine their level of familiarity with DevOps technologies. As a result, a group of respondents familiar with DevOps technologies was formed. The research identified and analyzed a list of DevOps technologies that have not yet been studied by future computer science teachers. The study employed a descriptive experiment using surveys and specialized software tools for data collection. Data analysis was conducted using descriptive statistics methods and the R programming language. The research aimed to understand respondents' attitudes toward DevOps technologies in the high school computer science curriculum, their views on specific DevOps technology elements for future computer science teachers, and the possible inclusion of DevOps technologies in higher education programs. To achieve these goals, a questionnaire with six questions was developed to assess the relevance of introducing DevOps technology elements into the "Secondary Education (Computer Science)" program. The survey was conducted in the autumn of 2022, targeting various educational stakeholders, including teachers, lecturers, school administrators, and extracurricular education institutions.

The survey results indicated that respondents believed collaborative work with data/documents, the use of coding in multiple fields, and cloud computing were highly relevant for the future development of society. Additionally, the research found that future computer science teachers should be proficient in software development technologies, cloud technologies/services, and managing various operating systems. Finally, respondents expressed a strong interest in adding DevOps technologies to the educational-professional program "Secondary Education (Computer Science)," with Infrastructure as Code, Containers, and Container Orchestration being the most desired DevOps technologies for inclusion.

5. Discussion:

The results of the conducted descriptive research on the issue of introducing the study of DevOps technology into the educational-professional program "Secondary Education (Computer Science)" have revealed significant interest among stakeholders.

It is pertinent to consider the survey results comprehensively. For example, the majority of respondents believe that the main directions of development in the information society are the use of coding (programming) elements in various fields and collaborative work with data/documents. These responses correlate with the answers to questions about proficiency in modern information technologies. This is evident in the selection of responses such as "software development technologies," "cloud technologies/services," and "technologies for managing various operating systems." Cloud services mostly provide users with tools for collaborative work with data and documents, and programming elements are indirectly used, even when creating and editing spreadsheets. The obtained results confirm the stated approaches to designing the educational-professional program "Secondary Education (Computer Science)" as discussed in N. Pavlova's research (Pavlova, 2022).

Analyzing the responses to the question "The preparation of a computer science teacher should meet the following requirements" allows us to conclude that the majority of stakeholders believe that a modern computer science teacher should be prepared to work with specialized computer science programs. In other words, they should teach computer science at the highest, specialized level. This is supported by 95% of the respondents.

The final question aimed to gather the opinions of respondents regarding which DevOps technologies should be taught to future computer science teachers. As noted in the research by N. Morze, T. Nanayeva, and O. Pasichnyk (Morze, Nanayeva & Pasichnyk, 2022), the high school computer science curriculum is overly theoretical, and teaching future teachers DevOps technologies would enable the introduction of a practice-oriented content into the high school computer science curriculum. The technologies selected by the respondents can be easily integrated into the educational-professional program "Secondary Education (Computer Science)" and can be used by graduates to enhance the practical orientation of computer science classes in general secondary education institutions when implementing specialized programs.

6. Conclusion:

The A modern computer science teacher in a general secondary education institution must be prepared for the challenges of today, the rapid evolution of information technologies, and their deep integration into all aspects of life. Content analysis of scientific research has revealed the issue of the absence of DevOps technology education in the preparation of future computer science teachers, while elements of this technology are increasingly permeating our lives. The conducted pedagogical exploratory experiment has confirmed the necessity of teaching DevOps technology to future computer science teachers as part of networking technology education. The experiment's results have allowed us to identify key topics in the field of DevOps (Infrastructure as Code, Containers, Container Orchestration) that are advisable to incorporate into the educational-professional program "Secondary Education (Computer Science)" for the training of future computer science teachers. Future research prospects include substantiating the improvement of the structure of educational components, taking into account the integration of the identified DevOps technologies into the educational-professional program "Secondary Education (Computer Science)," as well as studying the effectiveness of enhanced teaching content.

References

[1] Profile Level Educational Program for 10–11 Grades in Computer Science. (2011). Link Освітологічний дискурс, № 2(41), 2023 ISSN 2312-5829 (online)

[2] Pavlenko, M., Pavlenko, L., & Khomenko, V. (2019). Development of Application for Investigation and Testing in Python Educational Process. Physical and Mathematical Education, 4 (22), 100–107. Link

[3] Pavlenko, L. V., Pavlenko, M. P., Khomenko, V. H., & Mezhuyev, V. I. (2022). Application of R Programming Language in Learning Statistics. In Proceedings of the 1st Symposium on Advances in Educational Technology. 2, 62–72. Link

[4] Pavlova, N. (2022). Professional Training of Computer Science Teacher through the Educational and Professional Program of "Secondary Education (Computer Science)". Scientific Journal of M.P. Dragomanov National Pedagogical University. Series 5, Pedagogical Sciences: Realities and Perspectives, 88, 166–172. Link

[5] Morze, H., Nanaieva, T., & Pasichnyk, O. (2022). Teaching Computer Science in General Education in Ukraine. Information Technologies and Learning Tools, 92 (6), 1–20. Link

[6] Dyck, A., Penners, R., & Lichter, H. (2015, May). Towards Definitions for Release Engineering and DevOps. In 2015 IEEE/ACM 3rd International Workshop on Release Engineering, IEEE, 1–3. Link

[7] State of DevOps Report. (2021). Link

[8] Oehrlich, E., & Settle, M. (2022). Upskilling IT Annual Skills Report: Upskilling is a Professional and Organizational Imperative. DevOps Institute Inc. 1–58. Link

[9] Tamburri, D. A., & Perez-Palacin, D. (2018). DevOps Quality Engineering. Journal of Software: Evolution and Process, 1–3. Link

[10] Forsgren, N., & Humble, J. (2016). DevOps: Profiles in ITSM performance and contributing factors. In the Proceedings of the Western Decision Sciences Institute (WDSI). Link

[11] Vicente, A., & Cunha, J. (2022, June). Applying the DevOps Methodology for a More Efficient Process of Teaching-Learning Computer Programming. In 2022 31st Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE), IEEE, 1–6. Link

[12] Duvall, P. M., Matyas, S., & Glover, A. (2007). Continuous Integration: Improving software quality and reducing risk. Pearson Education.

[13] Cusumano, M. A., & Selby, R. W. (1998). Microsoft Secrets: How the world's most powerful software company creates technology, shapes markets, and manages people. Simon and Schuster.

[14] Forsgren, N., & Humble, J. (2016). The Role of Continuous Delivery in IT and Organizational Performance. In the Proceedings of the Western Decision Sciences Institute (WDSI). Link

[15] Mishra, A., & Otaiwi, Z. (2020). DevOps and Software Quality: A systematic mapping. Computer Science Review, 38. Link

[16] Лілія Павленко, Максим Павленко, Євген Павленко Educological discourse, 2023, Issue 2(41)

[17] Azad, N., & Hyrynsalmi, S. (2022, October). DevOps Challenges in Organizations: Through Professional Lens. In Software Business: 13th International Conference, ICSOB 2022, Bolzano, Italy, November 8–11, 2022, Proceedings, pp. 260–277, Cham: Springer International Publishing. Link

[18] Sánchez-Cifo, M. Á., Bermejo, P., & Navarro, E. (2023). DevOps: Is there a gap between education and industry? Journal of Software: Evolution and Process, e2534. Link

[19] Amaro, R. M. D., Pereira, R., & da Silva, M. M. (2022). Capabilities and Practices in DevOps: A multivocal literature review. IEEE Transactions on Software Engineering, 49 (2), 883–901. Link

[20] Lennon, R. G. (2022, August). DevOps Best Practices in Highly Regulated Industry. In Proceedings of Seventh International Congress on Information and Communication Technology: ICICT 2022, London, Volume 1, pp. 567–585, Singapore: Springer Nature Singapore. Link

[21] Best Upcoming DevOps Conferences in 2023. (2023). Eventyco. Link