Received 26 December 2024, accepted 27 August 2026, date of publication 30 September 2026
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.25.3.4852

Mathematical Model of DevOps Optimization: Balance between Development Speed, Stability, and Product Quality

MYKHAILO LUCHKEVYCH, IRYNA SHAKLEINA
Institute of Computer Sciences and Information Technologies,
Lviv Polytechnic National University, Lviv, 79013 Ukraine

Corresponding author: Mykhailo Luchkevych (e-mail: luchkevychmm@gmail.com).

ABSTRACT The concept of DevOps is considered an interdisciplinary approach that integrates software development and operational processes to improve the productivity and efficiency of IT teams. Literature analysis shows that although the practical aspects of DevOps are covered quite extensively, quantitative mathematical modeling of this process remains understudied. The focus is on DevOps evaluation metrics, key success factors, and the creation of a mathematical model that describes the relationship between parameters such as development speed, operational stability, automation, system complexity, and risk management.

The proposed model enables the optimization of DevOps processes while maintaining a balance between development speed and infrastructure stability. The impact of automation and CI/CD on overall software performance and quality is analyzed. Special attention is paid to challenges related to system complexity and resource efficiency. The research results are aimed at improving the theoretical foundations and practical tools for integrating DevOps in various industries.

KEYWORDS DevOps; mathematical modeling; process optimization; development speed; operational stability; automation; CI/CD.

I. INTRODUCTION

DevOps is an interdisciplinary concept that integrates software development and operations to optimize the creation, testing and deployment of software products [1-3]. This approach is aimed at reducing the delivery time of software solutions, improving their quality, and ensuring greater flexibility in adapting to changes in the business environment. Amid the ongoing digital transformation of organizations, DevOps is becoming increasingly important, as evidenced by the growing number of academic studies in this area [4-9].

A literature review shows that DevOps serves not only as a technological approach, but also as a methodology that actively transforms organizational processes. In particular, works [10-12] considered key aspects of the DevOps concept, including its definition, benefits, implementation challenges, and recommendations for implementing best practices. Studies [13-14] indicated that the application of DevOps principles significantly improves the efficiency of IT teams, increases customer satisfaction, and reduces operating costs. However, despite its obvious benefits, DevOps faces a number of serious challenges. Articles [15-16] emphasized such problems as resistance to changes in corporate culture, shortage of qualified specialists, and difficulty in implementing automated tools that are necessary to ensure continuous integration and delivery of software products.

Despite significant progress in developing practical approaches to implementing DevOps, academic research has largely focused on qualitative analysis of this process. Quantitative DevOps modeling, which could provide formalization of key parameters and creation of mathematical models to optimize its implementation, remains a less explored area. Such an approach could help identify dependencies between different aspects of DevOps, assess their impact on the overall performance of organizations, and develop scientifically based recommendations for process improvement. The development of this direction is important both for the practical implementation of DevOps and for the theoretical understanding of its role in the modern IT industry.

II. LITERATURE REVIEW

Article [17] provided a thorough overview of modern approaches to the use of metrics and Key Performance Indicators (KPIs) in the context of DevOps. The purpose of the

study is to systematize, analyze, and critically evaluate metrics used to determine the success of implementing DevOps practices in various organizations. The authors conducted a thorough analysis of the literature and identified 22 main DevOps metrics, including Mean Time to Recovery (MTTR), Deployment Frequency, Change Lead Time and Change Failure Rate.

The study also presented a categorization of metrics according to their functional characteristics, which allows for more effective application of these indicators in practical scenarios. For example, metrics were classified according to aspects of performance, reliability, process speed, and software quality. In addition, the article contains practical recommendations for integrating these metrics into DevOps processes, in particular taking into account the different stages of the software development and operation life cycle. The authors emphasize that implementing a metrics system can not only improve performance, but also facilitate more informed data-driven decisions, which in turn contributes to increased customer satisfaction and cost optimization.

Previous studies [8, 18-19] investigated critical success factors (CSFs) for DevOps projects. The analysis aimed to identify, structure, and evaluate factors that are key to effective DevOps implementation, as well as develop a framework model to optimize development processes and reduce deployment cycles. The authors identified three main categories of success factors: technical, organizational, and socio-cultural [8].

Technical factors include the availability of modern tools for process automation, the integration of configuration management systems, and an efficient software architecture that facilitates rapid deployment of changes. Organizational factors include proper planning, management support, resource allocation, and the implementation of systematic approaches to change management. Sociocultural factors include the team's willingness to change, openness to collaboration, and creating a culture of continuous learning.

The proposed CSFs framework for DevOps contributes to a deeper understanding of the key requirements for the successful implementation of this methodology. The authors emphasized that taking these factors into account not only increases the likelihood of success of DevOps projects, but also creates a sustainable environment for their further development. The study provided practical recommendations for developers, managers, and project leaders, including the adoption of best practices such as regular progress assessment, process transparency, and teamwork development. As a result, the works [8; 18-19] lay the foundation for future research in the field of DevOps management and the creation of more adaptive and effective models of process organization.

Article [18] proposed a model of Critical Success Factors (CSFs) for DevOps implementation, which covers four key areas: collaborative culture, DevOps practices, professional DevOps team, and metrics and measurements. A collaborative culture emphasizes the importance of creating an open environment that promotes effective interaction between development teams and operations professionals. DevOps practices include the use of techniques such as Continuous Integration (CI), Continuous Deployment (CD), and automated testing. A professional DevOps team emphasizes the need to form cross-functional teams with a high level of expertise. Metrics and measurements provide the ability to assess the effectiveness of DevOps implementation through key indicators that help identify problems and pinpoint areas for improvement.

The authors noted that the study has certain limitations due to the relative novelty of the DevOps concept. In particular, they recommended conducting further empirical research that takes into account industry-specific characteristics and provides a deeper understanding of how DevOps is implemented in different contexts.

Study [19] focuses on the relationship between organizational, technical, and socio-cultural factors that determine the success of DevOps. Organizational factors include effective internal communication, transparent hierarchy, and strategic planning. Technical aspects such as systems integration, test automation, and efficient infrastructure are the foundation for implementing DevOps practices. Sociocultural factors, including team dynamics and willingness to change, are crucial to creating a sustainable collaborative culture. The study highlights that these three groups of factors are closely interrelated and have a complex impact on the overall effectiveness of DevOps implementation. The authors concluded that for the successful implementation of DevOps, it is necessary to pay attention not only to technical tools and processes, but also to the cultural aspects of the organization and the development of team dynamics.

Article [20] demonstrates that implementing DevOps significantly improves software development, testing, and maintenance processes by introducing automation and integration between teams. The authors noted that three key components are important for achieving success in implementing DevOps: support from management, implementation of automated processes, and establishment of effective communication between teams. Leadership support facilitates strategic planning and resource provision, automated processes help reduce errors and speed up development cycles, and effective communication creates the foundation for harmonious interaction between teams. As a result, these elements contribute to increasing the efficiency and stability of business processes, which ensures the competitiveness of the organization in the market.

Overall, the generalization of research results [18-20] emphasizes the importance of a comprehensive approach to DevOps implementation, which takes into account technical, organizational, and socio-cultural aspects. Successful implementation of this approach requires strategic planning, leadership support, process automation, and the development of a collaborative culture that ensures long-term sustainability and effectiveness of organizations.

Article [21] proposed a structured representation of the DevOps concept in the form of a Venn diagram, which reflects the relationships between the main components of DevOps: software development, operational processes, and management practices. This diagram serves as a visual model for organizing the entities that influence the creation, support, and management of DevOps processes. The authors emphasize that this visualization not only clarifies the main components of DevOps but also identifies areas of intersection where collaboration between different teams is critical for success. This structural representation facilitates the selection of both theoretical and practical tools to formalize a single DevOps system model that can be applied to improve both existing processes and new initiatives.

An analysis of the available literature confirms that the conceptual and practical aspects of DevOps, including implementation methodologies, automation tools, principles of a collaborative culture, and change management, are widely researched and documented. However, as the authors noted, there is a significant gap in current research in the area of quantitative mathematical modeling of DevOps processes. The lack of clear mathematical models makes it difficult to accurately measure and predict the impact of key variables, such as development speed, degree of automation, operational stability, system architecture complexity, and risks, on the overall effectiveness of DevOps implementation.

Mathematical modeling could provide a new level of understanding for assessing DevOps effectiveness and optimization. For example, developing models based on system dynamics, queuing theory, or network analysis would allow us to identify dependencies between key parameters and assess the long-term impact of changes. Such models can become the basis for real-time decision-making when it is necessary to adapt processes to rapidly changing market conditions or technological requirements. In addition, quantitative modeling can contribute to the formation of an evidence base to justify DevOps implementation strategies in various industries, particularly in organizations with high operational complexity, such as the financial sector, healthcare, or telecommunications.

Thus, a structured presentation of the DevOps concept, as proposed in [21], combined with the development of quantitative mathematical modeling, opens up new perspectives for scientific research and practical application. These approaches can contribute to a more accurate understanding of the interaction mechanisms between key DevOps components and improve the efficiency of organizations in various sectors of the economy.

III. MATHEMATICAL MODEL

One of the key challenges in DevOps is balancing development speed with operational infrastructure stability. An excessively fast release pace can lead to reduced stability and lower product quality, while an overly conservative approach to operations can slow down the development and deployment of new features. At the same time, process automation and the implementation of CI/CD can reduce these risks, but their impact on the overall result depends on many factors, such as system complexity, resource availability, and risk management effectiveness.

A mathematical model is presented that describes the relationship between key DevOps parameters: development speed (D), operational stability (O), automation level (A), CI/CD implementation, and system complexity (C). The model provides a better understanding of how these variables affect the overall speed of software delivery and the quality of the final product, and provides tools for optimizing DevOps processes. In addition, the model takes into account resource and risk management, providing a more accurate basis for strategic decision-making in highly complex and dynamic environments.

The main variables of the model:

1) D (Development Speed) – the speed of development of new features and changes (measured, for example, in the number of features or releases over a certain period) [22];

2) O (Operational Stability) – stability and reliability of the operational infrastructure (measured by uptime, number of incidents or average service recovery time);

3) A (Automation Level) – the level of process automation, which can be measured depending on the percentage of processes that are automated [23];

4) CI/CD (Continuous Integration/Continuous Deployment Maturity) – the degree of implementation and use of CI/CD practices;

5) S (Delivery Speed) – the overall software delivery speed [24];

6) Q (Product Quality) – quality of the final product [25];

7) C (Complexity) – the complexity of the system, which can negatively affect the speed of development and operational stability [26];

8) R (Resources) – resources allocated for development, automation and support (human resources, infrastructure capacity);

9) Rsk (Risk Management) [27] – the level of risk management that can affect operational stability and product quality.

The speed of delivery depends on the efficiency of development processes, the level of automation and CI/CD, and taking into account the complexity of the system and resources:

$$S = \frac{D \times A \times CI/CD}{C} \times \left(1 + \frac{R}{R_{max}}\right), \quad (1)$$

where $\frac{R}{R_{max}}$ shows how efficiently resources are used relative to the maximum available level; C is the complexity factor, which reduces the overall delivery speed as the system complexity increases.

Quality depends on operational stability, automation, and risk management:

$$Q = O \times A \times (1 - R_{sk}), \quad (2)$$

where, R_{sk} - risks that may affect the quality (e.g., insufficient testing, technical debt), which should be minimized.

The balance between development speed (D) and operational stability (O) can be described as follows:

$$D + O = C + \frac{R}{R_{max}} + \alpha, \quad (3)$$

where α is the coefficient of external influences (e.g., changes in requirements, external constraints) that accounts for adjustments in the overall performance of the system.

The growth of D leads to a decrease in operational stability (O), but with a sufficient level of resources R and automation A, this impact can be minimized.

The DevOps model can be represented as an integration of all the above variables (1-3) to ensure speed and quality simultaneously:

$$\text{DevOps} = S \times Q = \left(\frac{D \times A \times CI/CD}{C} \times \left(1 + \frac{R}{R_{max}}\right)\right) \times \times (O \times A \times (1 - Rsk))$$

IV. RESULTS AND DISCUSSION

Interpretation of the model:

1) Impact of automation on DevOps effectiveness: Process automation (A) is a key factor that impacts both speed of delivery (S) and product quality (Q). The higher the level of automation, the more efficient DevOps is, as manual errors are reduced, risks are reduced, and the speed of deployment of new features increases;

2) Impact of system complexity: System complexity (C) negatively impacts delivery speed, as the increase in the number of components, services, and relationships slows down development, testing, and deployment. This requires careful management of the system architecture and its simplification when possible, or the expansion of automation to control complexity;

3) Balance between development speed and operational stability: The balance between development speed (D) and operational stability (O) is critically important. Increasing development speed at the expense of stability can lead to frequent incidents and a deterioration in product quality. The model shows that when one of these parameters increases, the other may suffer unless measures are taken to increase automation or additional resource involvement;

4) Resource efficiency: Resources (R) act as a compensating factor for the system. More human, technical, and financial resources can improve productivity, balance the impact of increasing complexity, and accelerate the adoption of automation, but effective use of these resources is essential to maximize results;

5) Risk Management: Risks (Rsk) play an important role in maintaining product quality. High levels of risk, including insufficient testing, technical debt, or frequent changes to requirements, can reduce product stability and quality, even if other variables are high. Risk management is a prerequisite for achieving stable and high-quality releases.

6) The Importance of CI/CD: The level of implementation and maturity of CI/CD processes directly impacts the speed of new software delivery. The more a company invests in CI/CD, the easier it is to integrate and deliver changes without disruption or risk, which improves the overall DevOps cycle;

7) Critical role of interactions between variables: The model shows that all variables are closely related to each other. Changing one component, such as development speed (D) or automation level (A), affects other aspects of the system, including quality and stability. To achieve effective DevOps practices, it is necessary to maintain a balance between these variables and ensure their optimal interaction.

Within the framework of the proposed mathematical model of DevOps optimization, each of the main variables has its own boundaries that characterize its range of values in the context of various aspects of development, automation, risk management, and operational stability. In Table 1, the numerical limits below define the minimum and maximum values for each variable.

Fig. 1 demonstrates the dependence of DevOps performance on two key factors: development speed (D) and operational stability (O). This graph helps visualize how the balance between development speed and operational stability affects overall DevOps performance. High operational stability with balanced development speed contributes to maximum productivity.

Table 1. Range of variable values

Variable	Minimum	Maximum
D (Development Speed)	0 releases or features per month. This value may indicate a lack of development of new features or releases due to technical or organizational constraints, or in phases of long-term refactoring.	30 releases or features per month, which corresponds to high-performance teams capable of releasing new features or releases daily.
O (Operational Stability)	80% uptime (service availability time). Low stability can indicate frequent failures and long recovery times, which significantly affects the reliability of the service.	99.999% uptime ("five nines"), which corresponds to the highest level of system stability with minimal incidents and prompt recovery.
A (Automation Level)	0% automation, which means a complete absence of automated processes, where all operations are performed manually.	100% automation, when all possible processes, including testing, deployment, monitoring, and other aspects of DevOps, are fully automated.
CI/CD	0%, indicating a lack of continuous integration and deployment practices, where processes are performed manually and irregularly.	100%, which reflects the full implementation of CI/CD practices, where all stages of integration, testing, and deployment are automated and performed continuously.
S (Delivery Speed)	0 releases per month, indicating a complete delay in releasing new versions or features, which may be the result of technical issues or delays in the development process.	30 releases per month (daily deployment of new versions or features), which corresponds to a highly efficient process with minimal delays.
Q (Product Quality)	<50% positive user feedback and more than 20 defects per release. This indicates a low-quality product with a large number of errors that need to be eliminated.	>99% positive user feedback and less than one defect per release, consistent with the highest quality product with minimal flaws and high user satisfaction.
C (Complexity)	1–3 components or services, which corresponds to a simple system with a limited number of interdependencies, making it easier to maintain and develop.	more than 100 components or microservices, which characterizes the high complexity of the system with a large number of interacting elements, which complicates its development, testing and support.
R (Resources)	1 person on the team and a minimum budget ($10,000 per year). This may indicate very limited resources, making it difficult to quickly develop and implement new features.	more than 50 people in the team and a significant budget (> $1,000,000 per year), enabling teams to scale projects and quickly implement innovations.
Rsk (Risk Management)	0%, indicating a lack of risk management processes where incidents are not assessed and controlled, leading to high risks of errors.	100%, which means full risk management, when all possible risks are assessed, monitored, and measures are taken to minimize them.

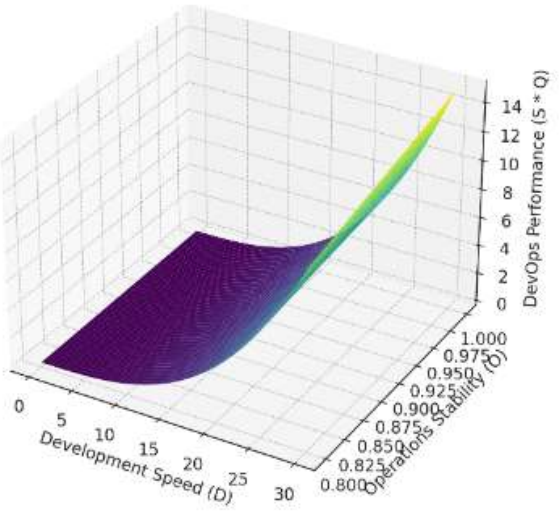

Figure 1. Dependence of DevOps performance on development speed (D) and operational stability (O)

Increasing the speed of development of new features or releases has a positive impact on DevOps productivity, but only up to a certain level. Excessive speed growth can negatively impact other aspects, such as operational stability, if there is no proper infrastructure support.

High stability of the operational infrastructure (approaching 100% uptime) has a positive impact on overall DevOps performance. If stability declines, performance drops accordingly, as unstable infrastructure leads to failures and longer recovery times.

Maximum DevOps performance is achieved when the balance between development speed and operational stability is maintained at an optimal level. The graph demonstrates that raising both D and O improves performance up to a certain point, beyond which there is a risk of losing stability due to excessive load on the operational infrastructure.

Fig. 1 shows that achieving high DevOps performance requires a carefully tuned balance between development speed and infrastructure stability, complemented by a sufficient level of automation and the implementation of CI/CD.

V. CONCLUSIONS

The technical, organizational, and socio-cultural factors that determine the success of DevOps projects are analyzed in this paper.

A mathematical model is presented that describes the relationships between the main DevOps parameters (development speed, operational stability, automation, CI/CD implementation, system complexity, resources, and risk management). The model makes it possible to determine optimal strategies for balancing these parameters.

It has been shown that the effectiveness of DevOps depends on maintaining a balance between development speed and operational infrastructure stability. Excessive emphasis on one of these parameters can negatively impact overall performance.

It has been shown that the level of process automation is a key factor that contributes to increasing the speed of software deployment, minimizing risks, and improving the quality of the final product.

It is emphasized that increasing system complexity requires careful management, including optimizing architecture and expanding automation.

The scientific novelty of the study is that the proposed model combines the software delivery speed S and the quality of the final product Q into a single quantitative structure. Due to this, the productivity of the DevOps team is considered not as a one-sided optimization of speed, but as the result of the simultaneous balancing of all related parameters of the model - from automation and CI/CD to system complexity, resources, and risk management. Such formalization and integration of interrelated DevOps parameters into a single optimization model constitute the main novel contribution of this study.

Prospects for further research are associated with empirical verification of the proposed model on data from real DevOps teams of different industry profiles and scales, which will allow us to refine the parameters and coefficients of the model, as well as assess the stability of the model and the sensitivity of its results to changes in input parameters. An important direction is calibrating the model on historical metrics from CI/CD platforms and monitoring systems, and refining it by detailing the risk management parameter Rsk. No less promising is the transition from a static relationship of parameters to an analysis of how changes in automation, complexity, or resource provision affect the trajectory of DevOps indicators throughout the project life cycle. Further development of the model may also include formalizing the socio-cultural factors identified in the study as important to DevOps success and integrating them into the proposed model as additional parameters. This will allow us to expand the model from a predominantly technical-organizational description to a comprehensive representation of the factors determining the effectiveness of DevOps teams.

References

[1] xC. Ebert, G. Gallardo, J. Hernantes and N. Serrano, "DevOps," *IEEE Software*, vol. 33, no. 3, pp. 94-100, 2016, https://doi.org/10.1109/MS.2016.68.

[2] A. Hemon-Hildgen, F. Rowe, "Conceptualising and defining DevOps: a review for understanding, not a framework for practitioners," *European Journal of Information Systems*, vol. 31, no. 5, pp. 568-574, 2022. https://doi.org/10.1080/0960085X.2022.2100061.

[3] R. T. Yarlagadda, "DevOps and its practices," *International Journal of Creative Research Thoughts (IJCRT)*, vol. 9, issue 3, pp. 2320-2882, 2021. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3798877.

[4] F. El Aouni, K. Moumane, A. Idri, M. Najib, S. U. Jan, "A systematic literature review on Agile, Cloud, and DevOps integration: Challenges, benefits," *Information and Software Technology*, vol. 177, 107569, 2025. https://doi.org/10.1016/j.infsof.2024.107569.

[5] R. Amaro, R. Pereira, M. Mira da Silva, "Mapping DevOps capabilities to the software life cycle: A systematic literature review," *Information and Software Technology*, vol. 177, 107583, 2025. https://doi.org/10.1016/j.infsof.2024.107583.

[6] J. Jayakody, W. Wijayanayake, "DevOps Maturity: A systematic literature review," *Proceedings of the IEEE International Research Conference on Smart Computing and Systems Engineering (SCSE)*, 2024, vol. 7, pp. 1-6, https://doi.org/10.1109/SCSE61872.2024.10550493.

[7] U. Hamza, N. L. Abdullah, S. M. Syed-Mohamad, "DevOps adoption guidelines, challenges and benefits: A systematic literature review," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 61, no. 4, pp. 114–136, 2024. https://doi.org/10.37934/araset.63.2.114136.

[8] N. Azad, S. Hyrynsalmi, "DevOps critical success factors – A systematic literature review," *Information and Software Technology*, vol. 157, 107150, 2023. https://doi.org/10.1016/j.infsof.2023.107150.

[9] J. Faustino et al., "DevOps benefits: A systematic literature review," *Software: Practice and Experience*, vol. 52, no. 9, pp. 1905-1926, 2022. https://doi.org/10.1002/spe.3096.

[10] N. Azad, S. Hyrynsalmi, "Multivocal literature review on DevOps critical success factors," *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering (EASE'24)*, 2024, pp. 520–527. https://doi.org/10.1145/3661167.3661236.

[11] R. Grande, A. Vizcaíno, F. O. García, "Is it worth adopting DevOps practices in Global Software Engineering? Possible challenges and benefits," *Computer Standards & Interfaces*, vol. 87, 103767, 2024. https://doi.org/10.1016/j.csi.2023.103767.

[12] J. Díaz et al., "Harmonizing DevOps taxonomies – A grounded theory study," *Journal of Systems and Software*, vol. 208, p. 111908, 2024. https://doi.org/10.1016/j.jss.2023.111908.

[13] M. Moeez, R. Mahmood, H. Asif, M. W. Iqbal, K. Hamid, U. Ali, N. Khan, "Comprehensive analysis of DevOps: Integration, automation, collaboration, and continuous delivery," *Bulletin of Business and Economics (BBE)*, vol. 13, issue 1, 2024. https://doi.org/10.61506/01.00253.

[14] U. Hamza, N. L. Abdullah, S. M. Syed-Mohamad, "DevOps adoption guidelines, challenges and benefits: A systematic literature review," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 61, issue 4, pp. 114-136, 2024. https://doi.org/10.37934/araset.63.2.114136.

[15] M. S. Khan et al., "Critical challenges to adopt DevOps culture in software organizations: A systematic review," *IEEE Access*, vol. 10, pp. 14339-14349, 2022. https://doi.org/10.1109/ACCESS.2022.3145970.

[16] L. Leite et al., "A survey of DevOps concepts and challenges," *ACM Computing Surveys*, vol. 52, no. 6, pp. 1-35, 2019. https://doi.org/10.1145/3359981.

[17] R. Amaro, R. Pereira, Mi. 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.

[18] V. Jayakody, J. Wijayanayake, "Critical success factors for DevOps adoption in information systems development," *International Journal of Information Systems and Project Management*, vol. 11, no. 3, pp. 60-82, 2023. https://doi.org/10.12821/ijispm110304.

[19] N. Azad, "Understanding DevOps critical success factors and organizational practices," *Proceedings of the 5th International Workshop on Software-intensive Business: Towards Sustainable Software Business*, 2022, pp. 83-90. https://doi.org/10.1145/3524614.3528627.

[20] A. Trigo, J. Varajão, L. Sousa, & D. T. Pham, "DevOps adoption: Insights from a large European Telco," *Cogent Engineering*, vol. 9, issue 1, 2022. https://doi.org/10.1080/23311916.2022.2083474.

[21] P.P. Maslianko, I.V. Savchuk, "DevOps – concept and structural representation," *KPI Science News*, no. 4, pp. 39-51, 2021. https://doi.org/10.20535/kpisn.2021.4.261938.

[22] A. Katal, V. Bajoria, S. Dahiya, "DevOps: Bridging the gap between Development and Operations," *Proceedings of the 3rd IEEE International Conference on Computing Methodologies and Communication (ICCMC)*, 2019, pp. 1-7, https://doi.org/10.1109/ICCMC.2019.8819631.

[23] M. A. W. Karunarathne, W. Wijayanayake, A. Prasadika, "DevOps adoption in software development organizations: A systematic literature review," *Proceedings of the 4th IEEE International Conference on Advanced Research in Computing (ICARC)*, 2024, pp. 282-287, https://doi.org/10.1109/ICARC61713.2024.10499789.

[24] N. N. Gadani, "Optimizing software development processes in cloud computing environments using agile methodologies and DevOps practices," *Asian Journal of Research in Computer Science*, vol. 17, no. 7, pp. 75-83, 2024. https://doi.org/10.9734/ajrcos/2024/v17i7479.

[25] D. Kato, H. Ishikawa, "Quality control methods using quality characteristics in development and operations," *Digital*, vol. 4, no. 1, pp. 232-243, 2024. https://doi.org/10.3390/digital4010012.

[26] H. R. Kadaskar, "Unleashing the power of DevOps in software development," *International Journal of Scientific Research in Modern Science and Technology*, vol. 3, no. 3, pp. 01-07, 2024. https://doi.org/10.59828/ijsrmst.v3i3.185.

[27] M. A. Akbar et al., "DevOps project management success factors: A decision-making framework," *Software: Practice and Experience*, vol. 54, no. 2, pp. 257-280, 2024. https://doi.org/10.3127/ajis.v28.4441.

***MYKHAILO LUCHKEVYCH**, Associate Professor of the Department of Information Systems and Networks Lviv Polytechnic National University, Associate Professor, Candidate of Sciences in Physics and Mathematics. Main research interests: distributed information systems, DevOps methodology and practice, data protection.*

***IRYNA SHAKLEINA**, Associate Professor of the Information Systems and Networks Department in Lviv Polytechnic National University, Candidate of Sciences in Physics and Mathematics. Main research interests: information technologies of intelligent data analysis and data engineering; machine learning; natural language processing; DevOps methodology and practice.*