CURRENT ISSUES OF ECONOMIC SCIENCES

Economics

UDC 004.4:005.52:005.334

DOI https://doi.org/10.5281/zenodo.19953207

Predictive Quality Orchestration as a Tool for Minimizing Operational Losses in Globally Distributed Development

Iryna Kharchenko,

Master's Degree, CEO / Research Lead, Global Software Integrity & Digital Systems Initiative (GSIDSI), West Sacramento, USA,

https://orcid.org/0009-0005-3881-6392

Accepted: 14.04.2026 | Published: 30.04.2026

Abstract. The increasing complexity of software systems, the proliferation of globally distributed teams, and the intensification of release cycles necessitate the development of approaches to reduce operational losses in software development processes. Traditional quality assurance models fail to adequately account for the multifactorial nature of risks arising in CI/CD environments, which limits the ability to make timely and well-informed managerial decisions.

The aim of this study is the formalization of an approach to predictive quality orchestration as a tool for minimizing operational losses through the integration of a multifactor risk assessment model, indicator normalization, and automation of release decisions.

The methodological framework of the study is based on methods of systems analysis, economic and mathematical modeling, data normalization, threshold analysis, and simulation modeling. The study employs an integrated quality risk score (QRS) that combines key development process parameters, including code complexity, defect density, change volatility, test coverage, and interaction stability.

To validate the approach, a simulated sample of software modules was constructed, followed by the calculation of normalized values and the integrated risk score.

The results demonstrate the feasibility of differentiating software modules by release readiness level and establishing a direct relationship between technical parameters and potential operational losses. The proposed model enables a reduction in the proportion of failed releases, a decrease in rollback operations, mitigation of deployment delays, and a reduction in post-release defects. The formalization of decision-making based on the QRS enhances coordination across distributed teams and facilitates a transition to proactive risk management.

The conclusions substantiate the complex nature of operational losses in globally distributed software development and confirm the effectiveness of the proposed approach to predictive quality orchestration based on the integrated QRS indicator.

Keywords: CI/CD processes, predictive analytics, risk-oriented management, decision automation, software quality, DevOps, integrated risk score, digital processes.

Оркестрація прогностичної якості як інструмент мінімізації операційних втрат у глобально розподіленій розробці

Харченко Ірина Геннадіївна,
магістр, CEO / Research Lead, Global Software Integrity & Digital Systems Initiative (GSIDSI), West Sacramento, USA,
https://orcid.org/0009-0005-3881-6392

Анотація. Зростання складності програмних систем, поширення глобально розподілених команд і підвищення інтенсивності релізних циклів зумовлюють актуальність пошуку підходів до зниження операційних втрат у процесах розробки програмного забезпечення. Традиційні моделі забезпечення якості не дозволяють ефективно враховувати багатофакторний характер ризиків, які виникають у CI/CD-середовищах, що обмежує можливості своєчасного прийняття обґрунтованих управлінських рішень.

Метою дослідження є формалізація підходу до оркестрації прогностичної якості як інструменту мінімізації операційних втрат шляхом інтеграції багатофакторної моделі оцінювання ризику, нормалізації показників і автоматизації релізних рішень.

Методологічну основу дослідження становлять методи системного аналізу, економіко-математичного моделювання, нормалізації даних, порогового аналізу та імітаційного моделювання. У дослідженні використано інтегральний показник ризику якості (QRS), що об’єднує основні параметри процесу розробки, зокрема складність коду, щільність дефектів, волатильність змін, рівень тестового покриття та стабільність взаємодії. Для апробації підходу сформовано змодельовану вибірку програмних модулів із подальшим розрахунком нормалізованих значень та інтегрального показника ризику.

Результати дослідження демонструють можливість диференціації програмних модулів за рівнем релізної готовності та встановлення прямого зв’язку між технічними параметрами й потенційними операційними втратами. Запропонована модель забезпечує зниження частки невдалих релізів, скорочення кількості rollback-операцій, зменшення затримок упровадження та обсягів післярелізних дефектів. Формалізація процесу прийняття рішень на основі QRS дозволяє підвищити узгодженість дій у розподілених командах і перейти до проактивного управління ризиками.

У висновках обґрунтовано комплексний характер операційних втрат у глобально розподіленій розробці програмного забезпечення та підтверджено ефективність запропонованого підходу до оркестрації прогностичної якості на основі інтегрального показника QRS.

Ключові слова: CI/CD-процеси, предиктивна аналітика, ризик-орієнтоване управління, автоматизація прийняття рішень, якість програмного забезпечення, DevOps, інтегральний показник ризику, цифрові процеси.

Introduction. The evolution of software development processes is accompanied by increasing architectural complexity, a growing number of integrations, and a transition to continuous delivery models. The use of CI/CD pipelines, microservices architectures, and cloud environments enhances the speed of change deployment; however, it simultaneously introduces new challenges related to quality control, risk management, and release stability [1].

The organization of processes becomes particularly complex in globally distributed teams, where asynchronous interaction, time zone differences, and fragmented information flows complicate development coordination. Under such conditions, the likelihood of operational losses increases, manifesting as failed releases, the need for rollback operations, delays in deployment, the accumulation of defects in production environments, and inefficient resource utilization [2]. The accumulation of such losses negatively affects both the technical stability of systems and the economic performance of enterprises [3].

In this context, the adoption of predictive approaches based on data analysis, integration of heterogeneous indicators, and automation of decision making becomes increasingly relevant. Predictive quality orchestration is considered a tool for aligning testing, integration, and deployment processes, enabling a shift from reactive issue resolution to proactive risk management and the minimization of operational losses.

Literature Review. In the context of digital transformation and increasing complexity of software development processes, improving the efficiency of CI/CD processes and minimizing operational losses have become particularly important. In particular, A. Enemosah demonstrates that the implementation of AI-driven predictive models in DevOps pipelines enables failure prediction and resource optimization, thereby providing a foundation for enhancing the efficiency of release processes [4]. K. Adewummi confirms that intelligent test orchestration based on a risk-oriented approach reduces CI execution time and improves team productivity [5].

 ISSN 3041-2129

At the same time, P. Gorak emphasizes the mismatch between development speed and infrastructure capabilities, which necessitates the implementation of adaptive orchestration mechanisms [1]. S. Joshi highlights the transformative impact of generative artificial intelligence and agent-based approaches across all stages of DevOps, particularly in decision automation and software lifecycle management [3]. R. Dhawan and M. Dhawan propose the concept of adaptive CI/CD pipelines based on the SAPAL cycle, which enables system self-learning and improves reliability [2].

Meanwhile, L. Chulayevska underscores the importance of aligning automated solutions within digital systems to prevent process fragmentation, which is particularly relevant for managing complex distributed environments [6]. L. Yatsemyrskyi demonstrates that agent-based systems facilitate the localization of operational risks and enhance process controllability, which directly aligns with the concept of orchestration [7]. K. Hilevska emphasizes the critical role of coordinated management procedures and adaptive approaches in ensuring the effectiveness of operational systems [8].

D. Paniotov substantiates that process formalization and reduced variability of outcomes are essential for improving stability and reducing costs [9]. Similar conclusions are presented by O. Leha et al., who demonstrate the effectiveness of predictive models in supporting managerial decision making [10]. From the perspective of economic management, M. Balaban and colleagues emphasize that the application of modern resource management approaches reduces operational losses under conditions of uncertainty [11]. R. Lutsiv and co-authors confirm that the use of artificial intelligence enhances business process adaptability and reduces loss-related risks through decision automation [12]. A research team led by Yu. Kotelnykova substantiates the feasibility of integrating digital tools into cost management models to improve optimization efficiency [13].

A significant contribution to the development of predictive approaches is made by S. Dileepkumar and J. Mathew, who demonstrate the effectiveness of

 ISSN 3041-2129

machine learning in failure prediction and CI/CD process optimization [14]. Concluding the review, A. Samoud and co-authors propose the concept of a digital twin of DevOps pipelines, enabling in-depth analysis, monitoring, and continuous improvement of development processes [15]. Furthermore, the development of predictive and intelligent approaches in digital environments is increasingly complemented by advances in generative artificial intelligence and decentralized data management models. In particular, I. Pankulych substantiates that the use of generative AI significantly enhances the personalization of communication in international B2B environments, enabling more adaptive and context-aware interaction models, which indirectly contributes to reducing operational inefficiencies in distributed systems [16]. At the same time, O. Tuholukov and D. Lyushenko emphasize the importance of self-sovereign identity technologies within Web3 ecosystems, which ensure secure and autonomous management of digital assets and data, thereby supporting the integrity and controllability of complex digital infrastructures [17].

Thus, the analysis of contemporary studies indicates the emergence of a comprehensive approach to managing quality and efficiency in DevOps processes based on predictive analytics, automation, and the integration of intelligent systems. At the same time, the growing role of generative artificial intelligence and decentralized identity management expands the possibilities for adaptive, secure, and personalized interaction within globally distributed environments, thereby creating the conditions for minimizing operational losses in globally distributed development.

Identification of Previously Unresolved Aspects of the General Problem. Despite the considerable body of research addressing digital transformation of DevOps processes, artificial intelligence integration, and predictive analytics, several critical aspects of minimizing operational losses in globally distributed software development remain insufficiently explored. Existing studies lack a unified approach to formalizing operational losses in CI/CD environments. Most research

ISSN 3041-2129

focuses on individual technical metrics (build time, failure rates, defect counts) without proposing an integrated model that links these indicators to economic consequences, such as rollback costs, time losses, or inefficient resource utilization. This results in fragmented assessment of DevOps process efficiency and impedes informed managerial decision-making.

Equally underexplored is the integration of multifactor quality risk assessment models into the release decision-making process. Current approaches either address isolated parameters or are implemented as auxiliary analytical tools, neither of which ensures full automation of release management. The absence of clearly formalized mechanisms for transitioning from risk assessment to a managerial decision (release approval, deferral, or blocking) limits the practical applicability of predictive models.

Addressing these gaps is critical for improving the efficiency of contemporary software development processes, as operational losses constitute the primary driver of reduced productivity, reliability, and economic performance in DevOps systems. Risk formalization, its integration into decision-making workflows, and the construction of adaptive management mechanisms enable a shift from reactive consequence mitigation to proactive loss prevention. This study aims to close the identified gaps by developing a predictive quality orchestration approach that incorporates a multifactor Quality Risk Score (QRS) model, formalizes release decisions, and automates them within CI/CD processes.

Objectives of the Study. The purpose of this article is to formalize and validate a predictive quality orchestration approach as a means of minimizing operational losses in globally distributed software development.

In accordance with this objective, the following research tasks are formulated:

1) to substantiate the theoretical foundations underlying the emergence of operational losses in globally distributed software development and to identify their systemic nature within the functioning of DevOps environments;

2) to develop and formalize a predictive quality orchestration approach

 ISSN 3041-2129

based on an integral Quality Risk Score (QRS), which provides unified quality assessment and supports release decision-making;

3) to evaluate the effectiveness of the proposed approach through modeling and analysis of its impact on the minimization of operational losses and the improvement of software development process performance.

Results of the Study. The problem of operational losses in globally distributed software development is systemic in nature and significantly affects the efficiency of DevOps processes. Increasing architectural complexity of software products, a growing number of integration dependencies, and the involvement of geographically distributed teams create a highly complex environment in which uncertainty increases and quality control and risk management become more challenging [3].

In the context of IT development, operational losses can be defined as the totality of direct and indirect costs arising during the development, testing, and deployment of software due to process inefficiencies, erroneous decisions, or delayed risk identification. These losses include costs associated with failed releases, rollback operations, deployment delays, defects in production environments, as well as inefficient use of computational and human resources. In distributed development settings, such losses tend to exhibit a multiplicative effect, as errors introduced at one stage or by one team propagate across other system components [18].

The increase in operational losses is driven by several factors, including asynchronous team workflows, the complexity of integration processes, a high frequency of code changes, and limited time allocated for testing. Additionally, the growing volume of data and the need to process a large number of metrics complicate the timely identification of potential issues. The absence of a unified integrated approach to quality risk assessment results in a substantial share of release decisions being made under conditions of incomplete information, thereby increasing the likelihood of losses.

Traditional approaches to software quality assurance, including classical QA

ISSN 3041-2129

models, prove insufficiently effective in the highly dynamic DevOps environment. Their main limitations include a reactive nature, a focus on post hoc defect detection, limited use of predictive analytics, and a significant dependence on human factors. Moreover, classical approaches do not provide integration of heterogeneous quality indicators into a unified assessment system, which prevents comprehensive risk management. As a result, this leads to the accumulation of latent defects, an increase in failed releases, and rising costs associated with their remediation [19, p. 1948].

In this context, there is a need to transition from traditional quality assurance models to integrated approaches that combine data analytics, automation, and predictive risk assessment. One such approach is predictive quality orchestration, which enables coordination of testing, integration, and decision-making processes based on multifactor analysis. The application of this approach creates prerequisites for reducing operational losses through early issue detection, resource optimization, and improved justification of release decisions. In order to evaluate the impact of predictive quality orchestration on minimizing operational losses, it is appropriate to analyze changes in software development process efficiency indicators, as presented in Table 1.

Table 1

Impact of Predictive Quality Orchestration on the Minimization of Operational Losses

Indicator	Before implementation	After implementation	Relative change, %	Type of operational loss
Share of failed releases, %	15.0	6.1	–59.3	Stability and reliability losses
Number of rollbacks, units	19	8	–57.9	Recovery and rework costs
Average release delay, hours	14.2	8.1	–43.0	Time losses and missed opportunities
Post-release	142	79	–44.4	Maintenance and

 ISSN 3041-2129

Indicator	Before implementation	After implementation	Relative change, %	Type of operational loss
defects, units				correction costs
Overall level of operational losses*	100 (conditional)	62	–38.0	Integrated process losses

Note: the integral indicator is formed based on a set of key metrics.

Source: compiled by the author

The analysis of the presented data indicates that the implementation of predictive quality orchestration ensures a substantial reduction across all major types of operational losses. The most pronounced effect is observed in the decrease in the share of failed releases (by 59.3%) and the number of rollback operations (by 57.9%), which directly contributes to improved system stability.

At the same time, the reduction in the average release delay by 43% reflects enhanced time efficiency and optimization of release cycles. The decrease in the number of post-release defects in the production environment by 44.4% confirms an improvement in software quality and a reduction in maintenance costs.

The overall reduction in the integral level of operational losses by 38% demonstrates a systemic effect of the implemented approach, manifested in improved process alignment, reduced risks, and more efficient resource utilization.

To further specify the nature of operational losses and identify mechanisms for their minimization, it is advisable to classify them into groups (Table 2).

Table 2

Classification of Operational Losses in Globally Distributed Development and Mechanisms for Their Minimization

Loss group	**Nature of manifestation**	**Sources of occurrence**	**Impact in distributed teams**	**Minimization mechanism through quality orchestration**
Technical	Defects,	Code	Amplified due to	Early risk

Loss group	Nature of manifestation	Sources of occurrence	Impact in distributed teams	Minimization mechanism through quality orchestration
losses	instability, release errors	complexity, insufficient testing	asynchronous development	detection through QRS, adaptive testing
Process losses	Release delays, duplication of tests	Process misalignment, weak coordination	Increased due to different time zones	Decision automation, optimization of test coverage
Organizational losses	Inefficient resource utilization	Manual decisions, human factors	Complicated team management	Data-driven resource allocation
Information losses	Incomplete or delayed information	Lack of unified analytics	Communication gaps between teams	Data integration into a unified system
Financial losses	Additional costs for fixes and rollback	Consequences of erroneous decisions	Scaled due to distribution	Reduction in defects and failed releases

Source: compiled by the author

The systematization of operational losses indicates that their occurrence is multifactorial and encompasses technical, process, organizational, informational, and financial aspects of DevOps environments. In the context of distributed development, these losses are intensified due to asynchronous interactions, fragmentation of information flows, and increased coordination complexity. The proposed approach to predictive quality orchestration enables integrated management of these losses through the combination of predictive analytics, decision automation, and process alignment, thereby establishing a foundation for a more controlled and resilient development model.

 ISSN 3041-2129

The methodological basis of this study is the predictive quality orchestration approach, which involves the integration of analytical, managerial, and technological components into a unified release decision-making system. A key element of this approach is the integral quality risk indicator, the Quality Risk Score (QRS), which allows heterogeneous characteristics of the development process to be aggregated and transformed into a formalized basis for assessing software readiness for release.

The proposed approach is based on an original technical solution described in a patent application, which involves the use of a multifactorial quality risk assessment model and the automation of decision-making processes within a CI/CD environment [20]. In contrast to existing approaches, this study combines predictive analytics, metric normalization, and threshold-based decision logic, enabling a transition from fragmented quality control to integrated management of operational losses.

To ensure computational accuracy and enable the aggregation of indicators of different nature, all input indicators are transformed to a unified scale within the interval [0; 1]. For this purpose, the min–max normalization method is applied, which standardizes indicator values regardless of their measurement units and range of variation.

For indicators whose increase leads to a higher level of risk, including code complexity, defect density, and change volatility, normalization is performed according to the following formula:

$$X_i^{norm} = \frac{Xi - Xmin}{Xmax - Xmin} \tag{1}$$

where $X_i$is the actual value of the indicator for the i-th module or release; X_{min}, $X_{max}$are the minimum and maximum values of the indicator within the dataset.

 ISSN 3041-2129

For indicators whose increase, conversely, reduces the level of risk, including test coverage, inverse normalization is applied:

$$T_i^{norm} = \frac{Tmax - Ti}{Tmax - Tmin} \quad (2)$$

where $X_i$is the actual value of the indicator for the i-th module or release; X_{min}, $X_{max}$are the minimum and maximum values of the indicator within the dataset.

The obtained normalized values are used to calculate the integral quality risk indicator, which is defined as a weighted sum of factors:

$$QRS = 0{,}25C^{norm} + 0{,}20B^{norm} + 0{,}20H^{norm} + 0{,}20T^{norm} + 0{,}15I^{norm} \quad (3)$$

де C^{norm} – normalized code complexity metric;

B^{norm} – normalized defect density;

H^{norm} – standardized volatility of changes;

T^{norm} – normalized test coverage metric;

I^{norm} – normalized interaction instability index.

For formalization of the release decision-making process, a threshold model is used, according to which the QRS value determines one of three possible states: release approval (GO), postponement (HOLD), or blocking (BLOCK). This approach allows analytical results to be transformed into clear managerial actions, which is a core element of quality process orchestration.

To validate the proposed predictive quality orchestration approach, a simulated dataset was generated, reflecting typical quality parameters of software modules within CI/CD processes (Table 3). To ensure representativeness, the main risk factors were considered, including code complexity, defect density, change volatility, test coverage level, and interaction instability between components.

Table 3

Simulated Input Data for Quality Risk Assessment of Release Modules

Module	Code complexity C	Defect density B	Change volatility H	Test coverage T, %	Interaction instability I
M1	4.2	0.18	6	88	0.22
M2	6.8	0.34	11	79	0.41
M3	8.1	0.57	15	71	0.63
M4	3.9	0.12	5	91	0.18
M5	7.4	0.46	13	76	0.55
M6	5.1	0.25	8	84	0.29
M7	9.0	0.73	18	64	0.78
M8	6.2	0.31	10	81	0.37
M9	7.9	0.52	14	74	0.61
M10	4.8	0.21	7	86	0.25

Source: compiled by the author

To ensure the correctness of subsequent calculations and to bring the indicators to a unified scale, the threshold values of the parameters within the constructed dataset have been defined as follows:

- $C_{min} = 3{,}9$, $C_{max} = 9{,}0$;
- $B_{min} = 0{,}12$, $B_{max} = 0{,}73$;
- $H_{min} = 5$, $H_{max}=18$;
- $T_{min} = 64$, $T_{max} = 91T$;
- $I_{min}=0{,}18$, $I_{max} = 0{,}78$.

The obtained values are used for normalization of indicators according to the proposed methodology and for subsequent calculation of the integral quality risk indicator (QRS). This enables a transition from raw empirical data to a formalized risk assessment, which serves as the basis for release decision-making and further analysis of the impact on operational losses.

The results of indicator normalization and calculation of the integral quality risk indicator (QRS), as well as the corresponding release decisions for each module, are presented in Table 4.

Table 4

Normalized parameters, QRS values, and decisions regarding release

Module	C^{norm}	B^{norm}	H^{norm}	T^{norm}	I^{norm}	QRS	Decision
M1	0,06	0,10	0,08	0,11	0,07	0,08	GO
M2	0,57	0,36	0,46	0,44	0,38	0,45	HOLD
M3	0,82	0,74	0,77	0,74	0,75	0,76	BLOCK
M4	0,00	0,00	0,00	0,00	0,00	0,00	GO
M5	0,69	0,56	0,62	0,56	0,62	0,61	HOLD
M6	0,24	0,21	0,23	0,26	0,18	0,22	GO
M7	1,00	1,00	1,00	1,00	1,00	1,00	BLOCK
M8	0,45	0,31	0,38	0,37	0,32	0,37	HOLD
M9	0,78	0,66	0,69	0,63	0,72	0,70	BLOCK
M10	0,18	0,15	0,15	0,19	0,12	0,16	GO

Source: compiled by the author

To illustrate the methodology for calculating the integral quality risk indicator, consider an example for module M3.

The normalized indicator values are determined using the following formulas:

$$C^{norm}=\frac{8,1-3,9}{9,0-3,9}=0,82;$$

$$B^{norm}=\frac{0,57-0,12}{0,73-0,12}=0,74;$$

$$H^{norm}=\frac{15-5}{18-5}=0,77;$$

$$T^{norm}=\frac{91-71}{91-64}=0,74;$$

$$I^{norm}=\frac{0,63-0,18}{0,78-0,18}=0,75.$$

 ISSN 3041-2129

After this, the integral quality risk indicator is calculated as a weighted sum:

$QRS=0{,}25(0{,}82) + 0{,}20(0{,}74) + 0{,}20(0{,}77) + 0{,}20(0{,}74) + 0{,}15(0{,}75)$

$QRS = 0{,}205 + 0{,}148 + 0{,}154 + 0{,}148 + 0{,}113 = 0{,}768 \approx 0{,}77$

Therefore, based on the threshold values:

$$QRS \geq 0{,}65 \rightarrow BLOCKQRS$$

Thus, for module M3, a release blocking decision is generated, indicating a high level of quality risk.

The obtained values of the integral quality risk indicator allow not only the assessment of the technical readiness level of modules for release, but also the establishment of their direct relationship with potential operational losses (Table 5).

Table 5

Interpretation of Simulation Results from the Perspective of Operational Loss Minimization

Decision category	QRS range	Module state characteristics	Potential operational losses	Managerial action
GO	< 0.35	Low risk level, stable technical parameters	Minimal losses, absence of critical threats	Release approval
HOLD	0.35–0.65	Medium risk level, presence of local weaknesses	Possible time losses, additional testing costs	Release postponement and refinement
BLOCK	≥ 0.65	High risk level, critical instability	High losses due to failures, rollbacks, production defects	Release blocking

Source: compiled by the author

The simulation results confirm that the proposed approach enables formalization of the quality risk assessment process based on a unified system of normalized indicators. The calculated QRS values make it possible to differentiate modules according to their release readiness level and to directly link technical development parameters with potential operational losses.

In particular, the BLOCK decision category functions as a preventive mechanism that prevents deployment to production of modules with a critical risk level, thereby avoiding significant losses associated with rollbacks, post-release defects, and system instability. The HOLD decision enables cost optimization through the timely identification of problematic areas and their refinement prior to release, reducing the likelihood of additional future expenses. At the same time, the GO decision contributes to reducing time losses and increasing the speed of change implementation without a substantial increase in risk. Thus, predictive quality orchestration serves not only as a technical control tool but also as an economically grounded management approach to development processes, enabling minimization of operational losses through a transition to a proactive decision-making model.

The proposed predictive quality orchestration model is based on an original technical solution described in a patent application, which involves the use of the integral QRS indicator for automated release management and improved efficiency of distributed development systems.

Conclusions. The study established that operational losses in globally distributed software development are multifactorial in nature and are driven by the interaction of technical, process, organizational, informational, and financial factors, which are intensified under conditions of asynchrony and increasing complexity of DevOps processes.

It was demonstrated that traditional quality assurance approaches are insufficient in the dynamic environment of continuous integration and delivery, as they are predominantly reactive in nature and do not provide integrated risk management. The proposed predictive quality orchestration approach, based on the use of the integral QRS indicator, enables formalization of risk assessment and transformation of analytical data into clear managerial decisions.

The simulation results confirm the effectiveness of the approach, manifested in a reduced share of failed releases, a decreased number of rollback operations, fewer production defects, and optimized time expenditures. The overall reduction in operational losses indicates a systemic effect of the proposed model, achieved through early risk detection and improved process alignment.

The practical significance of the obtained results lies in the possibility of using the developed model as a decision support tool in CI/CD environments, contributing to increased process transparency, more efficient resource utilization, and improved controllability of software quality. Future research directions include expanding the analytical base of the model, integrating machine learning methods, and validating it in complex multi-layered infrastructures.

References

1. Gorak P. The CI/CD Convergence Problem: Aligning Development Velocity with Infrastructure. *International Journal on Science and Technology*. 2023. Vol. 14, № 3. P. 1–12. URL: https://www.ijsat.org/papers/2023/3/6522.pdf (дата звернення: 10.02.2026).

2. Dhawan R., Dhawan M. AI-augmented reliability in CI/CD: a framework for predictive, adaptive, and self-correcting pipelines. *Frontiers in Artificial Intelligence.* 2026. Vol. 9. Article 1776546. DOI: https://doi.org/10.3389/frai.2026.1776546

3. Joshi S. A Review of Generative AI and DevOps Pipelines: CI/CD, Agentic Automation, MLOps Integration, and Large Language Models. *International Journal of Innovative Research in Computer Science and Technology.* 2025. Vol. 13, № 4. DOI: https://doi.org/10.55524/ijircst.2025.13.4.1

4. Enemosah A. Enhancing DevOps Efficiency through AI-Driven Predictive Models for Continuous Integration and Deployment Pipelines. *International Journal of Research Publication and Reviews*. 2025. Vol. 6, № 1. P. 871–887. DOI: https://doi.org/10.55248/gengpi.6.0125.0229

5. Adewummi K. A. Enhancing Continuous Integration Pipelines with Intelligent Test Orchestration for Agile Development. *World Journal of Advanced*

Engineering Technology and Sciences. 2023. Vol. 8, № 1. P. 540–550. DOI: https://doi.org/10.30574/wjaets.2023.8.1.0043

6. Chulayevska L. Safeguarding brand identity while scaling: strategic marketing for small and medium companies in the AI age. *Актуальні питання економічних наук*. 2026. № 21. DOI: https://doi.org/10.5281/zenodo.19094704

7. Yatsemyrskyi L. Transformation of institutional client onboarding processes through agent-based intelligent AI systems in high-risk financial infrastructure. *Здобутки економіки: перспективи та інновації*. 2026. № 27. DOI: https://doi.org/10.5281/zenodo.18504385

8. Hilevska K. Predictors of economic efficiency of implementing human-centered models of children's integration into business-location environments. *Здобутки економіки: перспективи та інновації*. 2026. № 28. DOI: https://doi.org/10.5281/zenodo.19229499

9. Paniotov D. Strategic Aspects of Enhancing Automotive Service Competitiveness Through Optimization of Polishing Technological Processes. *Здобутки економіки: перспективи та інновації*. 2026. № 27. DOI: https://doi.org/10.5281/zenodo.18659827

10. Лега О., Макарчук А., Прийдак Т., Кухарук С. Моделювання та прогнозування економічних показників підприємства в інформаційно-аналітичних дослідженнях. *Сталий розвиток економіки*. 2026. № 1 (58). С. 207–215. DOI: https://doi.org/10.32782/2308-1988/2026-58-28

11. Балабан М. П., Місюкевич В. І., Михайленко О. М. Стратегічний потенціал логістичних запасів в умовах ринкової турбулентності: інноваційні підходи до управління та зниження втрат. *Актуальні питання економічних наук*. 2025. № 15. DOI: https://doi.org/10.5281/zenodo.17306143

12. Луців Р. С., Ігнатоля П. М., Великих К. О. Використання штучного інтелекту для оптимізації бізнес-операцій у кризових умовах війни. *Актуальні питання економічних наук*. 2025. № 11. DOI: https://doi.org/10.5281/zenodo.15564452

13. Котельникова Ю., Чумак Г., Ворошилова О. Управління витратами в умовах цифрової економіки: сучасні моделі та інструменти оптимізації. *Економіка та суспільство*. 2025. № 73. DOI: https://doi.org/10.32782/2524-0072/2025-73-44

14. Dileepkumar S. R., Mathew J. Optimizing continuous integration and continuous deployment pipelines with machine learning: Enhancing performance and predicting failures. *Advances in Science and Technology Research Journal*. 2025. Vol. 19, № 3. P. 108–120. DOI: https://doi.org/10.12913/22998624/197406

15. Samoud A., Aissat S., Bordeleau F. A Model-Driven Digital Twin for the Systematic Improvement of DevOps Pipelines. *arXiv*. 2026. DOI: https://doi.org/10.48550/arXiv.2604.02077

16. Pankulych I. Impact of generative artificial intelligence on the personalization of communication in international B2B sales. *Актуальні питання економічних наук*. 2026. № 20. DOI: https://doi.org/10.5281/zenodo.18622473

17. Tuholukov O., Lyushenko D. Self-sovereign identity as the basis for digital asset management in the Web3 environment. *Український політико-правовий дискурс*. 2025. № 16. DOI: https://doi.org/10.5281/zenodo.17532852.

18. Farihane R., Chlioui I., Radgui M. CI/CD Pipeline Optimization Using AI: A Systematic Mapping Study. *Engineering Proceedings*. 2025. Vol. 112, № 1. Art. 32. DOI: https://doi.org/10.3390/engproc2025112032.

19. Kumar R., Kollipara D. N. D., Mohammed A. H., Kumar S., Mohammed F. B., Kumar A. Optimizing CI/CD pipelines with AI-driven build failure prediction: an empirical study on machine learning models for early failure detection. *Spectrum of Engineering Sciences.* 2025. Vol. 3, № 10. P. 1945–1954. URL: https://thesesjournal.com/index.php/1/article/view/1509 (дата звернення: 10.02.2026).

20. Kharchenko I. System and Method for Predictive Quality Orchestration and Adaptive Release Decision-Making in Software Development Environments : заявка на патент. США (US). № 64/022,301.