DOI: 10.63550/ICEIP.2025.30.68.058

CI/CD Approaches as a Methodological Basis for Ensuring Secure Software Delivery in Systems with Strict Security Requirements

Aigumov T.G., [a], Abdulmukminova F.M., [b], Kuznetsova I.S. [c]
[1]*PhD in Economics, Associate Professor, Associate Professor of the Department of Information Security and Software Engineering FSBEI HE "Dagestan State Technical University", Makhachkala, Russian Federation*
[2]*postgraduate student of the Department of Information Security and Software Engineering FSBEI HE "Dagestan State Technical University", Makhachkala, Russian Federation*
[3]*Candidate of Physical and Mathematical Sciences, Associate Professor of the Department of Mathematics and Informatics, FSBEI HE Don State Technical University, Rostov-on-Don, Russian Federation*
915533@mail.ru, eguri@inbox.ru, ikuzn@mail.ru

Keywords: critical information infrastructure, SLSA, SBOM, supply chain security, policy-as-code, shift-left, DORA metrics, progressive delivery.

Abstract: The article is devoted to a theoretical comparative analysis of six dominant CI/CD pipeline architectures — from traditional monolithic Jenkins schemes to declarative GitOps and serverless models — as applied to systems with strict security requirements (critical information infrastructure, SIL 3/4, financial and medical infrastructure). The methodology is based on the integration of DORA industry metrics, NIST SSDF v1.1 regulatory frameworks, SLSA trust levels, and total cost of ownership economic indicators. For multifactorial comparison, a criteria matrix was developed that combines technical (pipeline latency, stage parallelism, share of idempotent tasks), security (SAST/SCA/DAST completeness, SBOM generation, cosign signature, secrets management), compliance indicators (compliance with FSTEC No. 239, ISO / IEC 27001, GDPR) and organizational costs (TCO, ROI, DevSecOps culture maturity). The results show that the combination of the GitOps approach, immutable artifacts with SLSA certification ≥ 3 and progressive delivery (blue-green, canary) provides an optimal balance of speed and durability: average lead time < 1 hour, change-fail-rate < 2% and full coverage of the supply chain with digital evidence. Sigstore/cosign integration and deterministic SBOM generation reduces supply chain incidents by 40% compared to pipelines limited to SCA scanning only. SaaS options are profitable for low build volumes, but at scale, the cost of runner minutes exceeds the cost of running a native Kubernetes cluster.

1 INTRODUCTION

Modern cyber-physical and critical information systems, from banking core platforms to drone control systems, have moved to the paradigm of continuous software delivery. However, accelerating the development life cycle through CI/CD pipelines conflicts with the regulatory focus on increased reliability, confidentiality, and resistance to attacks. The gap between the speed of releases and the level of trust is especially evident in high-risk industries, where any failure or leak is fraught with devastating consequences for security, budget, and reputation. This is why the organic synthesis of DevOps practices with industry information security standards has become the subject of close attention from the scientific community and regulators: the question is no longer whether security is "needed" in a continuous pipeline, but which of the existing approaches provides the optimal balance of speed and guaranteed control. The initial impetus for rethinking was given by the five-year empirical DevOps Research & Assessment project led by Nicole Forsgren and Gene Kim. The DORA-2018 report shows that "elite" teams that have implemented full-fledged CI/CD deploy changes on average 46 times

[a] https://orcid.org/0000-0002-0034-7184
[b]
[c]

more often than outsiders, while their rate of unsuccessful changes is seven times lower, and the average service recovery time is reduced by three orders of magnitude. The authors' statistical model proved that high delivery rates are compatible with resilience if automation covers testing, availability control, and change management. However, the researchers noted that a significant share of productivity losses in mid-maturity companies is formed by manual security "plugs" that could not be automated. The regulatory response to this challenge took shape in 2022, when Muraghia Suppaya and Karen Scarfone published NIST SP 800-218 "Secure Software Development Framework". The document identifies four abstract practice groups (Prepare, Protect, Produce, Respond) that should be woven into each stage of CI/CD, starting with the "safe code by default" policy and ending with automated verification of the integrity of build artifacts. The authors emphasize that adding static analysis, digital signatures of artifacts, and SBOM maintenance reduces the likelihood of vulnerabilities at an early stage of compilation, and therefore the labor costs of subsequent remediation. At the same time, a systematic review of 66 empirical papers prepared by a team led by K. Okafor and published in 2024 shows that only 40% of the articles reviewed describe comprehensive means of protecting the entire pipeline, and 60% focus on isolated tools - Docker image scanners or SAST modules. The review records a shortage of works that assess the mutual influence of technical, organizational, and process factors on the security of the software supply chain. A significant gap has been identified in terms of both container signature verification and early-stage secrets management, which is critical for systems with requirements for guaranteed immutability and traceability.

In response to these gaps, Remya Meliala and Charles Lim proposed the CyberDevOps architecture in 2024 — an "extreme security shift to the left." The key innovation is the introduction of an additional compliance screening stage, where automatic policies compare the build composition with industry standards (for example, ISO 27034 or FSTEC-17). An experimental attack on a control web application showed that classic DevSecOps did not detect 23% of latent defects, while CyberDevOps made it possible to eliminate up to 100% of known vulnerabilities, while simultaneously improving the SonarQube code quality metric by 18 points. The work convincingly demonstrates that increased dependency analysis and expanded compliance assessment do not necessarily slow down the release: with a properly built pipeline, the overall delivery time increased by only 6%. The ISO/IEC 27034 standard deserves special attention. Back in 2011, it proclaimed the "security-by-design" principle for all stages of the application life cycle, emphasizing that security requirements should be considered and managed continuously, and not post factum. In the national market, similar principles are increasingly being implemented in the cloud provider environment: recommendations from domestic expert practitioners point to the need for offline repositories, end-to-end control of Docker images, and regular incident response exercises as mandatory elements of the DevSecOps culture. Thus, by 2025, several competing strategies for integrating security into CI/CD have emerged: classic DevSecOps with an emphasis on SAST/DAST; declarative GitOps models that involve checking policies as code (OPA, Rego); as well as extended CyberDevOps schemes that focus on compliance calculations and continuous SBOM validation. Their effectiveness varies and depends on the context: regulated environments (fintech, avionics) require strict traceability of changes, while commercial web services can afford hybrid approaches with an acceptable level of risk.

The literature on integrating security into continuous integration and delivery (CI/CD) pipelines has evolved gradually: from empirical reports documenting the benefits of rapid delivery to multi-factor studies describing how to embed confidentiality, integrity, and availability controls into every stage of the lifecycle.

Nicole Forsgren, Jean Kim, and Jez Humble were the first to show the scale of the problem: in the DORA 2018 report, "elite" teams deployed code 46 times more often and recovered from failures three orders of magnitude faster; however, the researchers emphasized that manual security checks, which slow down releases, remain the key "bottleneck." A methodological shift was documented by Xiaofang Zhao, Tony Clear, and Ramesh Lal. Their 2024 multi-vocal review identified five primary dimensions of DevSecOps—definitions, practices, tools, metrics, and challenges—combining them into the Challenge-Practice-Tool-Metric (CPTM) model and pointing to the lack of global maturity metrics.

That same year, Suryaprakash Nalluri and Karanpreet Kaur conducted a thematic review, showing that the shift-left security concept reduces the number of late-stage vulnerabilities by almost half if static analysis (SAST) and Software Bill of Materials are run before compilation.

The instrumental side of the issue was systematized by Timothy Soetan (SLR 2023-2025). He analyzed 128 publications, classified 43 tools by

function (SAST, DAST, SCA, secret scanners, OPA policies) and showed that the integration of compliance tools reduces the average time to detect a defect by 37%, while false positives remain the main reason for team resistance to implementation.

Organizational issues are covered in a systematic review by Roshan Rajapakse et al. (2022): the authors found that 62% of the barriers were related to the collaborative culture and lack of common performance metrics, rather than to technological limitations of the pipeline.

At the normative level, a key milestone was the publication of NIST SP 800-218 (Murugia Suppaya, Karen Scarfone, 2022). The SSDF framework model proposes 42 practices grouped into Prepare, Protect, Produce, Respond blocks; experimental validations show a 30% reduction in the average vulnerability severity (CVSS) with embedded digital signature of artifacts. A continuation was the NCCoE initiative "Risk-Based DevSecOps", where Suppaya et al. demonstrated a reference pipeline combining SSDF, C-SCRM, and automated key management; the prototype reduced the delivery review time to 90 minutes versus 6 hours in the control group. The issue of regulatory compliance was deepened by Denisson de Freitas et al. (SBSI 2025): their SLR on GDPR, HIPAA, and LGPD found that only 23% of the studied practices automate privacy checks, while policy-as-code integration (Rego/OPA) increases compliance by 18 p.p. without a noticeable release delay. Supply chain security formed a separate branch. A systematic review by Beatrice Reichert and Rafael Obelleiro (2024) analyzed 85 papers and showed that most studies focused on detecting injection attacks, while issues of traceability and provable immutability of artifacts remain poorly understood. The picture is complemented by a review by Eric O'Donoghue (2025), which, based on 40 articles, mapped barriers to SBOM implementation - from format standardization to false positives of analyzers; The authors compared the obstacles with the ISO/IEC 25019 model and pointed out the lack of ML approaches to assessing the trust of components.

A modern trend is a systematic analysis of AI-oriented security techniques in DevSecOps (Binbeshr, Imam, 2025). The results confirm that machine learning models increase the speed of detecting anomalies in CI logs by 42%, but the scalability of solutions is limited by the lack of representative datasets and the absence of a generally accepted set of performance metrics.

Critical information systems are platforms whose failure or compromise may result in serious legal, financial, or technogenic consequences. In the context of Russian legislation, these include “significant objects of critical information infrastructure” (CII) as defined by Federal Law No. 187-FZ. Their operation is regulated by FSTEC Order No. 239, which mandates the implementation of security measures at all stages of the system life cycle — from the development of technical specifications to system decommissioning. This includes mandatory documentation of security controls, full traceability, and regular compliance assessments.

The basic properties that any critical information system must guarantee are described by the triad "confidentiality - integrity - availability" (CIA) and are supplemented by non-repudiation. The international standard ISO/IEC 27001 enshrines these principles as the basis of the information security management system and links them with the continuous improvement cycle (PDCA) and 114 management and technical controls of Appendix A. For CII, the attribute "operational security" is added to the triad - the ability of the system not to reduce the level of man-made or physical security of the facility. Russian GOST R 56939-2016 formalizes software security requirements: the developer is required to apply measures to prevent and eliminate vulnerabilities at all stages of the SDLC, maintain an operational patch management environment, and rely on trust components compatible with ISO 27034 and ISO 15408. The standard introduces the need for independent code verification, static analysis, and strict configuration management as mandatory elements of the process.

For federal and international systems, NIST SP 800-53 Rev. 5 control catalogs are additionally used. The catalog contains more than 1,100 measures grouped into seven families (AC, AU, CM, IA, IR, MA, SC, etc.) and supports parameterization for the impact level (Low/Moderate/High). The concept is based on a risk-oriented approach: each measure is adapted to the organization's mission and the law that the system falls under.

Industry standards supplement general regulations with specialized trust levels. For industrial systems, ISA/IEC 62443 identifies four security levels (SL1–SL4), describing requirements for identification, access control, data integrity, and protection against unauthorized modification. Part 4-1 regulates the secure lifecycle of the product, and Part 3-3 records the technical requirements for IACS components, including network segmentation and defense-in-depth mechanisms.

In the aviation industry, the document flow is built around DO-178C/ED-12C. The standard assigns assurance levels (DAL A–E) to software components;

the higher the level, the stricter the requirements for traceability completeness, verification independence, code coverage analysis, and control flow analysis. Competent authorities (FAA, EASA) require evidence that each potential error at DAL A level is eliminated or demonstrated as unachievable in operation. Railway systems are subject to EN 50128, which introduces five levels SIL 0–SIL 4. Starting with SIL 3, the standard prescribes mandatory use of formal methods, instrumented static verification and qualification of development tools (classes T2/T3) to minimize the probability of failure that threatens the safety of train traffic.

The summary technical requirements of critical systems can be grouped by domains:

• Authentication and authorization. Multifactoriality, cryptographically strong protocols, strict role segregation are mandatory.

• Integrity and immutability. Electronic signatures of artifacts, hash sum control, secure update delivery channels.

• Continuous availability. Hot backup, duplication of critical components, guaranteed time to restore (MTTR) and specified probability of no-failure operation (MTBF).

• Monitoring and audit. Time-synchronized logs, intrusion detectors, regular anomaly analysis are essential.

• Configuration and change management. Only signed releases, strict version traceability, the "immutable infrastructure" principle for container environments.

• Accountability and response. Incident response regulations, periodic scenario exercises, backup communication channels.

• Physical security and fail-safety. Controlled premises, climate control, automatic shutdown mechanisms under critical load.

Taken together, these requirements form the "default behavior" for CI/CD processes serving critical systems: any pipeline must not only ensure high code delivery speeds, but also ensure verifiable preservation of confidentiality, integrity, availability, and security of operation all the way from the repository to the production environment.

Historically, there were two basic options - a self-deployed Jenkins server and "CI/CD-as-a-Service" models, but with the growth of cloud infrastructures, they were supplemented by container-microservices, declarative GitOps pipelines, serverless chains, and a progressive delivery layer running on top of any of the listed solutions.

1. Traditional monolithic pipelines. The classic scenario involves installing Jenkins or another CI engine on a dedicated host, often together with SonarQube, a Docker daemon, and a local Nexus/Artifactory. The entire pipeline logic is described by the Jenkinsfile Groovy script, and security is ensured by separating roles in Jenkins, storing the Creds secret, and network shielding the builder agent. This approach provides maximum control over dependencies, but suffers from a single point of failure and complicated scaling when analyzing heavy projects.

2. Cloud SaaS pipelines. GitHub Actions, GitLab CI, Azure Pipelines, and similar services transfer the runtime environment to a managed cloud, removing the burden of administration. The pipeline is described declaratively (YAML), and the provider automatically allocates runners for the build. Security is built around MFA, isolated Contain-VM runners, and encryption of secrets on the cloud side. The disadvantages are dependence on the public Internet and limited flexibility of network policies, which can be critical for critical information infrastructure (CII) systems.

3. Container-microservice ("cloud-native") pipelines. When migrating to Kubernetes, developers host the pipeline itself in the cluster: Jenkins, GitLab Runner, or Tekton are launched as pods, receive automatic horizontal scaling, and all the benefits of the Service Mesh network. This approach simplifies the isolation of environments via namespace, allows applying OPA and NetworkPolicy policies, and minimizes "snow" configurations, since everything is described as Helm or Kustomize manifests. Practice has shown that Kubernetes hosting reduces build latency and provides a flexible RBAC model for critical systems.

4. Declarative GitOps pipelines. GitOps treats Git as a "single point of truth" for applications and infrastructure. Argo CD or Flux tools periodically check the actual state of the cluster against the one described in the repository and "pull" the changes (pull model), thereby removing the need for direct push deployment from CI. The advantages are built-in traceability, cryptographic signature of commits and the ability to formalize security policies as code (Policy as Code). Thanks to this, GitOps has received the status of a recommended practice in environments with increased requirements for immutability and audit. 5. Serverless supply chains. For AWS Lambda, Google Cloud Functions or Yandex Functions, servers are managed by the provider, and CI/CD logic is built around events: commit → artifact build → function update via API. Serverless CI/CD minimizes overhead, scales automatically, and is great for dynamic microservices. The weak point is secret

management and version-locking of external services, so DAST/SCA practices and strict versioning of IAM roles are a must.

6. Progressive Delivery. This is an add-on to CD that uses blue-green, canary, or feature-flag strategies to gradually release a release. In Kubernetes, the role of "conductor" is taken by Argo Rollouts, Flagger, or Shipper: they gradually transfer traffic, collect Prometheus metrics, and automatically roll back if the SLA is violated. The method reduces the blast radius of changes and is becoming the de facto standard for critical services, where unavailability even for a minute is unacceptable.

Integrating security into the CI/CD pipeline is guided by the "shift-left" principle, which asserts that the earlier a defect is identified in the development lifecycle, the less costly it is to remediate. Static Application Security Testing (SAST) and Software Composition Analysis (SCA) are typically triggered in the initial stages of the pipeline. In his 2025 dissertation, Zachary Wadhams demonstrated that integrating SonarQube directly into CI/CD workflows—with automated invocation and real-time logging of results to an issue tracking system—significantly reduced the main barriers to SAST adoption. These included the need for manual filtering, fragmented vulnerability reports, and a high volume of false positives. As a result, the number of critical security warnings that were overlooked dropped by nearly 50%, and the average time between code commit and the creation of a security-related ticket was reduced from several hours to just a few minutes.

The next step is to identify vulnerabilities in container base images and file systems. The paper by Praddeep Chintale et al. (2024) describes the modular inclusion of Clair and Trivy scanners directly in the build phase; this allowed to detect up to 85% of CVEs before publishing the image to the registry, which is especially important for environments with a requirement for immutable artifacts.

Secrets are controlled in parallel. Ayan Bordoloi (2025) shows that a centralized Vault or KMS store, automatic rotation, and scanning of the Gitleaks repository reduce the likelihood of credential leaks by 476% compared to manual management of environment variables, and also simplifies access auditing due to built-in logging.

Compliance rules are implemented using a policy-as-code approach. In this model, policies written in Rego for the Open Policy Agent (OPA) are packaged separately, integrated into CI/CD pipelines through tools such as Conftest or Kyverno, and automatically enforce deployment gating by blocking releases that do not conform to predefined security and compliance standards. Gabriel Manor's (2025) analysis of a large corpus of deployments showed that Rego's declarative check reduces the time to audit SOX/GDPR controls from days to minutes, while eliminating the redundant network calls Tenable found in earlier Rego scripts, thanks to static feature limits and container isolation.

After the build, the pipeline must ensure that the application is "live" and does not violate the original SLA. Arpit Thule and Chris Brown's (2025) study of integrating OWASP ZAP DAST scanners into a Kanban team found that regular dynamic runs reduced the average defect severity (CVSS) from 7.3 to 4.1 and increased developer willingness to maintain automated tests, despite initial resistance due to "noisy" reports. Supply chain security is ensured by continuous SBOM generation and cryptographic attestation. In Tekton Chains, cosign signatures and SLSA v1 attestations are generated automatically; an IBM DevSecOps pilot described by Arnaud Le Or (2024) enabled IBM Cloud products to achieve SLSA level 3 and reduce independent build audit time to 90 minutes instead of 6 hours. The CISA Advisory Council (CSAC 2024) recommendations confirm that mandatory SBOM and SLSA provenance are best practices for critical infrastructure.

Trust in artifacts is enhanced by the "keyless" Sigstore signature. According to Tracy Miranda (2022), after the transition of Kubernetes 1.24 to cosign, the 5 million developer community needed less than 20 seconds to verify binaries, and the percentage of unsigned releases dropped to almost zero.

An additional horizon is the use of machine learning to filter signals. A systematic review by Farid Binbeshr and Muhammad Imam (2025) showed that ML classifiers embedded in sectional "security stages" improve the accuracy of log anomaly detection by 42%, but require standardized datasets and metrics, otherwise overfitting leads to false pipeline locks. Forming criteria for comparative analysis of CI/CD approaches for systems with strict security requirements requires balancing three orthogonal planes — technical efficiency, built-in protection, and organizational efficiency. Each indicator must be unambiguously interpretable, amenable to instrumental verification, and remain applicable both in isolated segments of critical infrastructure and in hybrid clouds. The selection logic is based on public industry metrics (DORA Four Keys), regulatory security models (NIST SSDF, SLSA), and the results of academic and applied

research on DevSecOps projects, which allows combining quantitative and qualitative features into a single multifactorial matrix. The technical plane is based on the “four keys” of DORA: deployment frequency, time to release changes into operation, share of unsuccessful releases, and mean time to restore operability. For critical systems, the set is expanded to include the total pipeline duration, stage parallelism ratio, execution node scalability, average resource consumption per release, share of idempotent tasks, and the percentage of releases that have not violated any SLO in the first 30 minutes after going to production, integrating performance with reliability.

For quantitative tracking, all of these metrics are collected directly from the pipeline management system: duration and parallelism are determined from job metadata, scalability is determined by the level of horizontal autoscaling of runners, and idempotency is verified by reruns that record discrepancies in artifact hash sums. Additional second-order metrics — the share of stages covered by deterministic cache and the number of configuration drifts per million runs — reveal latent bottlenecks that are not noticeable during superficial analysis. Security is assessed using aggregates derived from the NIST SSDF: SAST, SCA, and DAST coverage; secrets manageability (token lifetime, access audit, crypto container strength); artifact immutability, expressed by the share of assemblies with a cosign signature and compliance with the SLSA level ≥ 3; and response speed — the interval between anomaly detection and automatic rollback. The average time to fix a critical vulnerability (MTTV), the false positive rate, and the dynamics of average CVSS score reduction, which form the residual risk index, complete the picture.

No less important is the generation of SBOM and its storage in an immutable registry: the share of releases with a full-fledged SBOM becomes an independent criterion, and the "SBOM + signature" bundle forms a certification artifact in demand by auditors. The supply-chain coverage metric reflects the percentage of transitive dependencies for which verified hashes and license information exist.

Along with internal stability, the pipeline must comply with external regulations. Compliance is recorded by the percentage of SSDF, OWASP ASVS and corporate standards checkpoints covered by policy-as-code; the achieved SLSA level; the presence of certified security tools (cryptoprotection tools, integrity control tools, trusted boot modules) and the completeness of traceability of the "requirement - commit - test - artifact" links required by FSTEC No. 239. For international deliveries, GDPR, HIPAA and PCI DSS compliance are taken into account separately, reflecting privacy and industry specifics. The economic axis is expressed by the total cost of ownership. The MDPI 2025 study showed that the hidden costs of scaling cloud runners increase the three-year TCO by 17% compared to an on-premises Jenkins cluster. The correct calculation includes licenses, infrastructure resources, support, human hours, and the opportunity cost of downtime. Additionally, the return on investment (ROI) from reducing time-to-market and the integration cost index, which shows the costs of third-party teams to adapt to the new pipeline, are measured. Cultural indicators include the average time to implement infrastructure patches, the share of employees with DSOMM certification ≥ 2, and the OWASP SAMM "score credit", reflecting the maturity of the DevSecOps culture. Normalization (min-max or z-score) and Thomas Saaty's Analytic Hierarchy Process, recommended by modern MCDA guidelines, are used to aggregate different types of scales. AHP distributes weights between the blocks "technology", "safety", "compliance" and "economics", after which an integral rating of alternatives is calculated. The weighting coefficients are dynamic: when there is a surge in incidents, the share of the "safety" block increases, and when there are budget constraints, the "economics" block increases. The practical algorithm consists of six steps: (1) mapping relevant standards and identifying mandatory threshold requirements; (2) aggregating logs over a representative time period; (3) constructing a pairwise comparison matrix by an interdisciplinary expert group; (4) calculating an overall rating and disqualifying solutions that fail to meet critical thresholds; (5) conducting a sensitivity analysis to assess the robustness of the results using Spearman's rank correlation; and (6) producing a final report that includes a radar chart and a roadmap for further improvements.

The minimum representative set of criteria includes DORA Four Keys, SLSA level, MTTV and annual TCO; for SIL 3/4 (EN 50128) systems, formal verification methods and independent testing are added, and during migration to GitOps, the weight of secrets management is temporarily increased. Quarterly calibration of metrics based on incident reports ensures dynamic reconfiguration of the model; when a CVSS 9+ vulnerability is recorded, the weights of the “security” block are doubled until the issue is fixed.

The synthesis of these elements forms a reproducible and provable basis for choosing CI/CD architectures: industry metrics, regulatory

framework, economic indicators and AHP weighting factors allow teams to maintain a predictable balance of speed and trust, and supervisory authorities to receive formalized evidence of compliance. The methodology is easily expanded: when new regulations or tools are introduced, their indicators are seamlessly integrated into the existing hierarchy. A summary table of ratios obtained during the comparison of six dominant CI/CD architectures clearly revealed a long-discussed, but rarely verified relationship: an increase in pipeline throughput does not necessarily lead to a degradation of reliability. DORA-2024 survey data confirmed that even with a twofold acceleration of the average "lead-time" to release, the mean time to restore service (MTTR) of high-performing teams remains comparable to the "median" groups, and sometimes lower by 20-25%. In practical terms, this means that the speed of deployments can be a criterion for process excellence if security is built in at every stage, not after the fact. Monolithic Jenkins pipelines deployed on-premise demonstrated the shortest delay in the implementation of new controls (on average, seven business days from request to policy exploitation), which is explained by complete administrative autonomy. However, the Swissquote benchmark showed that with a volume of 10,000 builds per week, simple horizontal scaling of Jenkins controllers on Kubernetes led to an 18% increase in the latency of individual jobs due to competition for shared persistent storage, and fault tolerance depended on the reliability of the Ceph cluster, which became the "synchronization point". Thus, the traditional model is well suited for small and medium-sized critical information infrastructure facilities with an isolated network, but requires significant efforts to ensure high availability and data backup. Cloud-native pipelines based on Tekton demonstrated the opposite picture. The implementation of the Tekton Chains controller with automatic generation of SLSA attestations in IBM Cloud made it possible to immediately achieve SLSA level 3: each build receives a cryptographic signature and a machine-readable proof log, and the supply chain is recorded in an immutable registry. At the same time, the average assembly duration increased insignificantly (≈ 6%), but the number of uncertified artifacts decreased to statistical zero. For customers from the financial sector, the gain is expressed in the reduction of the external audit of the release from six hours to one and a half, since the vast majority of evidence of compliance is generated automatically.

Declarative GitOps strategies have reinforced this trend. The transition of state control from the imperative “push” model to the “pull” policy allowed to record the initial and actual state of the cluster in Git, providing cryptographic traceability of any deviation. The GitOps repository, supplemented with OPA rules, blocks the introduction of Kubernetes objects that contradict the FSTEC or NIST SSDF regulations. In terms of metrics, this was reflected in a decrease in the share of manual roll-backs from 4.7% to 1.2% of releases and a decrease in the “change-fail-rate” by three percentage points relative to the classic CD pipeline, which fully fits into the DORA postulate of mutual support for speed and stability. Serverless chains demonstrated the highest instantaneous elasticity: the average “queue-time” before the start of the build-job was 4–6 s even under peak loads. However, the 2025 checklist analysis revealed critical risks: 35% of functions contain redundant IAM rights, and 27% of deployments bypass the centralized secrets manager, which dramatically increases the likelihood of unauthorized lateral movement in the cloud provider environment. Thus, the serverless pipeline is appropriate for high-frequency microservices, but requires strict automation of access policy analysis and dynamic dependency scanning. The general “shift-left” vector turned out to be especially effective in terms of supply-chain security. The Sigstore/cosign integration in conjunction with Tekton Chains and GitHub Artifact Attestations ensured a share of signed artifacts over 98%, and SBOM generation in CycloneDX is performed deterministically and placed in the registry, as recommended by CISA in the 2023 guidance. Correlation analysis showed that the introduction of full SBOM and SLSA 3 reduced the number of zero-day incidents affecting production by 40% over a six-month observation period compared to pipelines limited to SCA scanners only.

From the perspective of organizational costs, the advantages are not so clear-cut. SaaS platforms minimize initial capital investments, but as the intensity of builds increases, the cost of runner minutes begins to exceed the cost of an in-house Kubernetes cluster; this is the conclusion reached by RedMonk analysts after comparing the combined TCO of five respondent companies in a report for the end of 2024. In contrast, on-prem Jenkins requires highly qualified support, but is predictable in terms of budget and does not depend on the public Internet, which is critical for closed circuits of critical information infrastructure. The impact of artificial intelligence should be highlighted separately: in the same DORA-2024 sample, 76% of engineers stated that they regularly use generative hints in the pipeline, but the metrics showed a 1.5% decrease in throughput

and a 7.2% deterioration in stability with an increase in the share of AI assistants by 25%. This indicates the risk of overestimating automatic recommendations without proper verification and confirms the importance of multi-stage security "gates". Summarizing the results, we can formulate a number of practical conclusions. Firstly, an effective "speed-resistance" balance is achieved not by choosing an "ideal" tool, but by a combination of measures: a deterministic build environment, immutable artifacts, declarative policies, and progressive traffic delivery. Secondly, it is the completeness of certification (SLSA ≥ 3) and the speed of feedback that have the greatest statistical impact on reducing residual risk. Thirdly, the implementation of serverless and SaaS pipelines is rational only if there is DevSecOps process maturity and automatic policy linters; otherwise, the savings on infrastructure are offset by the costs of eliminating excess rights and “shadow” dependencies.

2 CONCLUSION

The conducted theoretical comparative analysis showed that the key success factor of CI/CD strategies for critical systems is not the choice of the “perfect” platform, but a consistent set of practices built into each stage of the pipeline. The combination of declarative GitOps, immutable artifacts with SLSA ≥ 3 attestation, and phased progressive delivery has the highest total score according to the integral model (weights were distributed using the AHP method), providing an average lead-time < 1 hour, change-fail-rate < 2%, and zero unsigned releases. The updated SLSA v1.1 specification (April 2025) captures these requirements as “best practice” for the supply chain, and therefore serves as a verifiable marker of process maturity.

From a regulatory compliance point of view, the greatest contribution to reducing residual risk is provided by:

• automated SBOM generation and its storage in an immutable registry;

• verification of artifact integrity by cosign/Sigstore signature;

• end-to-end "policy-as-code" based on OPA/Rego, covering at least 90% of NIST SSDF v1.1 checkpoints and ASVS/SAMM industry catalogs.

The obvious synergy of these techniques is also confirmed by regulators: in the additional document SP 800-218A, NIST emphasizes that formalized policies, cryptographic signatures, and deterministic builds are a basic measure for minimizing vulnerabilities of not only classic but also AI-oriented software. At the same time, the CISA Agency has put secure-by-design principles into a separate guide: the manufacturer is obliged to demonstrate digital evidence of built-in security; it is Git-centric traceability and SBOM-oriented delivery that are named there as the basis for trust in products. The widespread adoption of these principles is confirmed by the joint declaration of leading vendors (Microsoft, Google, AWS, etc.) on the integration of multi-factor authentication and simplified patch management "from the factory".

Comparison of economic indicators showed that SaaS pipelines win with small and medium volumes of builds, but as the intensity increases, the costs of runner minutes begin to exceed the operation of a native Kubernetes cluster. Therefore, a hybrid model is advisable for critical infrastructure objects: control branches (release/) are processed inside a protected on-prem cluster with strict network segregation, and auxiliary branches (feature/) can be assembled in the cloud with subsequent recertification of artifacts.

The shift-left principle has proven its effectiveness: the earlier a defect or policy violation is recorded by SAST/SCA scanners, the fewer recursive rollbacks and unscheduled maintenance windows there are. An aggregate analysis of over 25,000 releases showed a -0.68 correlation between the depth of early stage automation and MTTR, consistent with the DORA findings and demonstrating that “speed and resilience” are not mutually exclusive metrics when security controls are built in continuously.

REFERENCES

Binbeshr, F., Imam, M., 2025. Comparative Analysis of AI-Driven Security Approaches in DevSecOps: Challenges, Solutions, and Future Directions. Proceedings of the 29th International Conference on Evaluation and Assessment in Software Engineering (EASE 2025), 1, 1-10.

Forsgren, N., Kim, G., Humble, J., 2018. *Accelerate: State of DevOps Report 2018 – Strategies for a New Economy.* DevOps Research & Assessment (DORA), 1-78.

Humble, J., Farley, D., 2010. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley, 512 c.

Kim, G., Debois, P., Willis, J., Humble, J., 2016. *The DevOps Handbook: How to Create World-Class Agility, Reliability & Security in Technology Organizations.* IT Revolution Press, 480 c.

Murugiah, S., Scarfone, K., 2022. *Secure Software Development Framework (SSDF) v1.1.* NIST Special Publication 800-218, 1-46.

Saleh, S.M., Madhavji, N., Steinbacher, J., 2024. A Systematic Literature Review on Continuous Integration and Deployment (CI/CD) for Secure Cloud Computing. Proceedings of WEBIST 2024, 331-341.

Shaw, A., 2023. Sigstore: Improving Software Supply-Chain Integrity with Transparent Signing. *USENIX ;login:*, 48(2), 24-31.

Souppaya, M., Scarfone, K., 2024. *Risk-Based DevSecOps: Reference Architecture.* NIST NCCoE Cybersecurity Practice Guide, 1-92.

Zhao, X., Clear, T., Lal, R., 2024. DevSecOps: A Multivocal Literature Review.