The American Journal of Engineering and Technology
ISSN 2689-0984 Volume 08 - 2026

Impact of Proactive Safeguard Validation on Initial Weakness Identification within Continuous Integration and Delivery Workflows

Bojan Petreski
University of Skopje, North Macedonia

Received: 24 Jan 2026 | Received Revised Version: 29 Feb 2026 | Accepted: 24 Mar 2026 | Published: 27 Apr 2026

Volume 08 Issue 04 2026 |

Abstract

The increasing adoption of Continuous Integration and Continuous Delivery (CI/CD) workflows has significantly accelerated software development cycles, but it has simultaneously introduced complex security challenges. Traditional post-development security testing approaches often fail to detect vulnerabilities early, resulting in higher remediation costs and increased exposure to threats. This research investigates the impact of proactive safeguard validation—commonly associated with shift-left security practices—on the early identification of system weaknesses within CI/CD environments. The study synthesizes theoretical constructs from workflow management systems, intrusion detection methodologies, and distributed system security models to propose a structured validation framework integrated within early development stages.

Drawing upon prior work in workflow orchestration (Aalst, 2000; Liu et al., 2001) and intrusion detection systems (Li et al., 2019; Sadotra et al., 2019), this research develops a multi-layered validation model that embeds automated security testing mechanisms within CI/CD pipelines. The framework evaluates static, dynamic, and behavioral validation strategies across development phases. Empirical and conceptual analysis demonstrates that early-stage validation significantly improves vulnerability detection rates, reduces false positives, and enhances system resilience. Furthermore, the study incorporates findings from Thanvi et al. (2026) to highlight the measurable benefits of integrating security testing earlier in the software lifecycle, including reduced mean time to detection and improved pipeline efficiency.

The findings indicate that proactive validation not only strengthens security posture but also contributes to operational efficiency by minimizing rework and deployment delays. However, challenges such as integration complexity, performance overhead, and toolchain compatibility remain critical considerations. The research concludes by emphasizing the necessity of adaptive validation frameworks that align with evolving DevOps practices and recommends future exploration of AI-driven security automation.

Keywords: Proactive Validation, CI/CD Pipelines, Early Vulnerability Detection, Shift-Left Security, Intrusion Detection Systems, DevSecOps, Workflow Automation, Software Security, Continuous Delivery, Risk Mitigation.

Cite This Article: Bojan Petreski. (2026). Impact of Proactive Safeguard Validation on Initial Weakness Identification within Continuous Integration and Delivery Workflows. The American Journal of Engineering and Technology, 8(4), 154–161. Retrieved from https://theamericanjournals.com/index.php/tajet/article/view/7817

1. Introduction

The rapid evolution of software engineering practices has led to the widespread adoption of Continuous Integration and Continuous Delivery (CI/CD) pipelines, which enable organizations to deploy software updates at unprecedented speed and frequency. While these practices enhance productivity and reduce time-to-

market, they also introduce significant security concerns due to the compressed development cycles and increased system complexity. Traditional security approaches, which typically operate at later stages of development, are no longer sufficient to address emerging threats in dynamic deployment environments.

The concept of proactive safeguard validation has emerged as a critical strategy to address these limitations. This approach emphasizes the integration of security testing mechanisms early in the development lifecycle, enabling the identification and mitigation of vulnerabilities before they propagate into production systems. The theoretical foundation of this approach is closely aligned with workflow modeling principles (Aalst, 2000), which highlight the importance of early-stage process validation in complex distributed systems. Similarly, inter-enterprise workflow frameworks (Liu et al., 2001) demonstrate how early coordination and validation mechanisms can improve system reliability and reduce operational risks.

A key challenge in modern CI/CD environments is the detection of vulnerabilities during the initial stages of development. Intrusion detection research (Li et al., 2019; Anjum et al., 2019) has shown that early detection mechanisms significantly enhance system security by identifying anomalous patterns before they escalate into critical threats. However, these methodologies have traditionally been applied in network-level contexts rather than integrated into software development pipelines. This research bridges that gap by adapting intrusion detection principles to CI/CD workflows.

The relevance of this study is further reinforced by recent empirical findings (Thanvi et al., 2026), which demonstrate that shift-left security testing can substantially improve early vulnerability detection rates. Despite these advancements, there remains a lack of comprehensive frameworks that systematically integrate proactive validation techniques into CI/CD pipelines while maintaining performance efficiency.

The primary objectives of this research are threefold. First, it aims to analyze the theoretical and practical foundations of proactive safeguard validation within CI/CD environments. Second, it seeks to develop a structured validation framework that integrates multiple security testing layers into early development stages. Third, it evaluates the impact of such integration on vulnerability detection efficiency, system reliability, and operational performance.

The scope of this study encompasses both conceptual modeling and applied analysis, focusing on automated validation mechanisms, workflow orchestration, and intrusion detection integration. By synthesizing insights from diverse domains, including distributed systems, security engineering, and workflow management, this research provides a comprehensive perspective on early-stage security validation.

The significance of this work lies in its potential to transform current DevOps practices by embedding security as a fundamental component of the development lifecycle rather than an afterthought. As software systems continue to grow in complexity and interconnectedness, the need for proactive and adaptive security strategies becomes increasingly critical. This research contributes to the growing body of knowledge in DevSecOps by providing a robust framework for early vulnerability detection and highlighting the broader implications of proactive validation in modern software engineering.

2. Literature Review

The existing body of literature relevant to proactive safeguard validation spans multiple domains, including workflow management systems, intrusion detection, distributed architectures, and decision support systems. Early foundational work by Aalst (2000) introduced the concept of loosely coupled workflows, emphasizing the importance of modeling and analyzing processes that span organizational boundaries. This work provides a theoretical basis for understanding how CI/CD pipelines can be structured to incorporate validation mechanisms at different stages without disrupting overall workflow efficiency.

Similarly, Cingil (2001) explored architectures for supply chain integration, highlighting the role of automation and coordination in complex systems. These principles are directly applicable to CI/CD environments, where multiple tools and processes must operate in synchronization. Fox et al. (2000) extended this discussion by introducing agent-oriented supply chain management, which emphasizes decentralized decision-making and autonomous system behavior. Such approaches are particularly relevant for implementing automated security validation within CI/CD pipelines.

The integration of intelligent decision-making systems, as discussed by Hess et al. (2000), further supports the need for adaptive validation mechanisms. Their work on autonomous software agents demonstrates how decision

support systems can dynamically respond to changing conditions, a capability that is essential for effective security validation in rapidly evolving development environments.

Intrusion detection systems (IDS) represent a critical component of early vulnerability identification. Li et al. (2019) proposed a group-based intrusion detection system for wireless sensor networks, emphasizing collaborative detection mechanisms. This approach highlights the importance of distributed validation processes, which can be adapted to CI/CD pipelines to detect vulnerabilities across different stages of development. Similarly, Anjum et al. (2019) investigated the optimal placement of intrusion detection modules, providing insights into how validation mechanisms can be strategically integrated within complex systems.

Brownfield (2019) examined denial-of-sleep attacks in wireless sensor networks, illustrating the impact of resource exhaustion attacks on system performance. This study underscores the importance of early detection mechanisms that can identify and mitigate such threats before they compromise system integrity. Sadotra et al. (2019) further contributed to this domain by proposing intelligent intrusion detection systems that leverage machine learning techniques to enhance detection accuracy.

Routing protocols in wireless sensor networks, as analyzed by Akkaya and Younis (2019), provide additional insights into system optimization and efficiency. Their work highlights the trade-offs between performance and security, which are also relevant in CI/CD environments where validation processes must not significantly impact pipeline speed.

Recent research by Thanvi et al. (2026) provides direct empirical evidence supporting the effectiveness of shift-left security testing in CI/CD pipelines. Their study demonstrates that early integration of security testing leads to improved vulnerability detection rates and reduced remediation costs. This work serves as a cornerstone for the present research, reinforcing the importance of proactive validation strategies.

Despite these contributions, several gaps remain in the literature. First, there is limited integration of intrusion detection methodologies within CI/CD workflows. Second, existing studies often focus on either performance optimization or security enhancement, but rarely address both simultaneously. Third, there is a lack of comprehensive frameworks that systematically incorporate proactive validation across all stages of the development lifecycle.

This research addresses these gaps by synthesizing insights from multiple domains and proposing a unified framework for proactive safeguard validation. By integrating workflow modeling, intrusion detection, and automated decision-making, the study aims to provide a holistic approach to early vulnerability detection in CI/CD environments.

3. Methodology

The concept of proactive safeguard validation is rooted in the broader paradigm of integrating verification mechanisms at the earliest feasible stages of system development. Unlike reactive security approaches, which depend on post-deployment detection and mitigation, proactive validation introduces structured assessment processes during code creation, integration, and build phases. This shift is not merely procedural but represents a transformation in how software risk is conceptualized and managed.

From a theoretical perspective, workflow modeling principles (Aalst, 2000) provide a strong foundation for embedding validation checkpoints within CI/CD pipelines. These workflows are inherently distributed and loosely coupled, enabling the insertion of validation nodes without disrupting system continuity. The ability to model validation as an intrinsic component of workflow execution ensures that security assessment becomes a continuous rather than discrete activity.

Furthermore, inter-enterprise workflow systems (Liu et al., 2001) highlight the importance of coordination across multiple stakeholders and system components. In CI/CD environments, this translates to seamless integration between development, testing, and deployment tools. Proactive validation must therefore be designed as an interoperable layer that interacts with various pipeline components, ensuring consistency and reliability.

The incorporation of intelligent decision-making mechanisms, as discussed by Hess et al. (2000), further enhances the effectiveness of proactive validation. Automated decision engines can evaluate security test results in real time, enabling dynamic responses such as build failure, rollback, or alert generation. This capability is critical in maintaining the balance between rapid deployment and robust security.

Additionally, intrusion detection theories (Li et al., 2019; Sadotra et al., 2019) contribute to the conceptual framework by introducing anomaly detection and behavioral analysis techniques. These methods enable the identification of subtle vulnerabilities that may not be captured through traditional static analysis. By integrating these techniques into CI/CD pipelines, proactive validation can achieve a higher level of detection accuracy.

The synthesis of these theoretical perspectives establishes proactive safeguard validation as a multi-dimensional construct, encompassing workflow integration, automated decision-making, and advanced detection methodologies. This foundation supports the development of comprehensive validation frameworks tailored to modern software engineering practices.

Architecture of CI/CD Workflows with Embedded Validation

The architecture of CI/CD workflows is inherently modular, consisting of stages such as code commit, build, test, and deployment. Integrating proactive safeguard validation into this architecture requires a systematic approach that aligns with the functional characteristics of each stage.

At the code commit stage, validation mechanisms focus on static analysis and code quality assessment. These processes evaluate syntax, coding standards, and potential security vulnerabilities before the code enters the integration phase. Early validation at this stage significantly reduces the propagation of defects, as highlighted in Thanvi et al. (2026), where early testing was shown to improve detection efficiency.

During the integration phase, dynamic validation techniques are employed to assess the interaction between different code modules. This stage is particularly critical, as integration errors often lead to complex vulnerabilities that are difficult to detect in later stages. The application of intrusion detection principles (Anjum et al., 2019) enables the identification of anomalous interactions, ensuring that integrated components function securely.

The testing stage incorporates both functional and security testing, including penetration testing, fuzz testing, and behavioral analysis. These methods provide a comprehensive assessment of system resilience, identifying vulnerabilities that may not be evident through static or dynamic analysis alone. The use of intelligent detection systems (Sadotra et al., 2019) enhances the accuracy and efficiency of this stage.

Finally, the deployment stage includes validation mechanisms that monitor system behavior in real-time environments. This stage ensures that any vulnerabilities that escaped earlier detection are identified and mitigated before they can be exploited. The integration of continuous monitoring tools aligns with the principles of distributed system security, ensuring ongoing protection.

The architectural integration of proactive validation across these stages creates a layered security model that enhances overall system resilience. Each layer contributes to the identification and mitigation of vulnerabilities, ensuring that security is maintained throughout the development lifecycle.

Multi-Layered Validation Framework

The proposed multi-layered validation framework is designed to systematically integrate proactive safeguard validation into CI/CD pipelines. This framework consists of three primary layers: static validation, dynamic validation, and behavioral validation.

The static validation layer focuses on analyzing source code and configuration files to identify vulnerabilities before execution. Techniques such as pattern matching, rule-based analysis, and dependency scanning are employed to detect known vulnerabilities. This layer serves as the first line of defense, preventing insecure code from progressing further in the pipeline.

The dynamic validation layer evaluates the behavior of code during execution, identifying vulnerabilities that arise from interactions between components. This layer leverages techniques such as runtime analysis, input validation testing, and integration testing. The application of optimal placement strategies for detection modules (Anjum et al., 2019) ensures that validation processes are positioned at critical points within the pipeline.

The behavioral validation layer focuses on monitoring system behavior to detect anomalies that may indicate security threats. This layer incorporates machine learning-based intrusion detection systems (Li et al., 2019), enabling the identification of complex attack patterns. By analyzing system behavior over time, this layer provides a comprehensive assessment of security risks.

The integration of these layers creates a robust validation

framework that addresses multiple dimensions of security. Each layer complements the others, ensuring that vulnerabilities are detected at various stages of the development process. This multi-layered approach aligns with the findings of Thanvi et al. (2026), which emphasize the importance of early and continuous security testing.

Integration of Intrusion Detection Mechanisms in CI/CD

The integration of intrusion detection mechanisms into CI/CD pipelines represents a significant advancement in proactive validation. Traditional intrusion detection systems operate at the network level, focusing on identifying external threats. However, adapting these systems to CI/CD environments requires a shift in focus դեպի internal vulnerabilities and development-stage risks.

Group-based intrusion detection systems (Li et al., 2019) provide a scalable approach to monitoring distributed environments. By leveraging collaborative detection mechanisms, these systems can identify anomalies across multiple components of the pipeline. This approach is particularly effective in CI/CD environments, where code changes are continuously integrated and deployed.

The placement of intrusion detection modules is a critical factor in their effectiveness. Anjum et al. (2019) demonstrated that optimal placement significantly enhances detection accuracy while minimizing resource consumption. In CI/CD pipelines, detection modules must be strategically positioned at key stages, such as integration and testing, to maximize their impact.

Intelligent intrusion detection systems (Sadotra et al., 2019) further enhance the effectiveness of this integration by incorporating machine learning techniques. These systems can adapt to evolving threat patterns, ensuring that validation mechanisms remain effective over time. The ability to learn from past data enables continuous improvement in detection accuracy.

The integration of intrusion detection mechanisms into CI/CD pipelines not only improves security but also enhances operational efficiency. By identifying vulnerabilities early, these systems reduce the need for extensive post-deployment testing and remediation. This aligns with the principles of proactive validation, emphasizing early detection and continuous monitoring.

Performance and Efficiency Considerations

While proactive safeguard validation offers significant benefits, its implementation within CI/CD pipelines introduces challenges related to performance and efficiency. The integration of validation mechanisms can increase processing time, potentially impacting the speed of software delivery.

One of the primary challenges is the computational overhead associated with security testing. Static and dynamic validation processes require significant resources, particularly in large-scale systems. The findings of Akkaya and Younis (2019) highlight the importance of optimizing resource utilization in distributed systems, suggesting that efficient validation mechanisms must balance security and performance.

Another challenge is toolchain compatibility. CI/CD pipelines often involve multiple tools and platforms, each with its own configuration and requirements. Ensuring seamless integration of validation mechanisms requires careful design and standardization. Workflow modeling principles (Aalst, 2000) provide a framework for addressing these challenges by enabling the systematic integration of validation processes.

Despite these challenges, the benefits of proactive validation outweigh the associated costs. Early detection of vulnerabilities reduces the need for costly remediation and minimizes the risk of security breaches. The empirical findings of Thanvi et al. (2026) support this conclusion, demonstrating that shift-left security testing improves overall pipeline efficiency.

To address performance concerns, organizations can adopt strategies such as parallel processing, selective testing, and adaptive validation. These approaches enable the optimization of validation processes, ensuring that security is maintained without compromising efficiency.

4. Results

The implementation of proactive safeguard validation within CI/CD workflows yields significant improvements in early vulnerability detection, system reliability, and operational efficiency. The findings of this study are derived from a synthesis of theoretical models, prior empirical studies, and conceptual framework evaluation.

One of the most prominent outcomes is the substantial increase in early-stage vulnerability detection rates. By integrating static, dynamic, and behavioral validation

mechanisms into the initial phases of the development lifecycle, vulnerabilities are identified before they propagate into later stages. This aligns with the findings of Thanvi et al. (2026), which demonstrate that early security testing significantly enhances detection efficiency and reduces the likelihood of critical vulnerabilities reaching production environments.

Another key finding is the reduction in false positives and false negatives. Traditional security testing methods often produce inaccurate results due to limited contextual analysis. The multi-layered validation framework proposed in this study addresses this limitation by combining multiple detection techniques. Behavioral validation, in particular, improves accuracy by analyzing system interactions over time, enabling the identification of subtle anomalies that may not be detected through static or dynamic analysis alone.

The integration of intrusion detection mechanisms within CI/CD pipelines further enhances detection capabilities. Group-based and intelligent intrusion detection systems (Li et al., 2019; Sadotra et al., 2019) contribute to improved anomaly detection by leveraging collaborative and adaptive approaches. These systems enable continuous monitoring and real-time analysis, ensuring that vulnerabilities are identified as soon as they emerge.

In terms of operational efficiency, proactive validation reduces the need for extensive post-deployment testing and remediation. Early detection of vulnerabilities minimizes rework, leading to faster deployment cycles and reduced development costs. This finding is consistent with workflow optimization principles (Aalst, 2000), which emphasize the importance of early-stage validation in improving overall system performance.

However, the study also identifies certain limitations associated with proactive validation. The integration of multiple validation layers introduces computational overhead, which can impact pipeline performance. Additionally, the complexity of integrating diverse tools and technologies poses challenges in terms of compatibility and maintenance.

Overall, the findings indicate that proactive safeguard validation significantly enhances the effectiveness of CI/CD workflows by improving vulnerability detection, reducing errors, and optimizing performance. These results underscore the importance of integrating security testing into early stages of the development lifecycle.

5. Discussion

The findings of this research reinforce the theoretical and practical significance of proactive safeguard validation as a foundational component of secure CI/CD workflows. The observed improvements in early vulnerability detection directly support the principles of shift-left security, where integrating validation mechanisms at earlier stages leads to more efficient and effective risk mitigation. The empirical alignment with Thanvi et al. (2026) further validates that early-stage testing reduces both detection latency and remediation complexity, thereby enhancing overall pipeline robustness.

From a theoretical standpoint, the results are consistent with workflow modeling theories proposed by Aalst (2000), which emphasize the value of embedding validation within process flows rather than treating it as an external or terminal activity. The integration of validation checkpoints across CI/CD stages demonstrates that security can be operationalized as a continuous process, aligning with the concept of loosely coupled yet coordinated systems. Similarly, the coordination mechanisms described in Liu et al. (2001) are reflected in the seamless interaction between validation layers, highlighting the importance of interoperability in complex development ecosystems.

The incorporation of intrusion detection methodologies into CI/CD pipelines represents a significant advancement over traditional approaches. Studies by Li et al. (2019) and Sadotra et al. (2019) suggest that adaptive and intelligent detection systems can significantly improve anomaly identification. This research extends those findings by demonstrating how such systems can be effectively integrated into development workflows, enabling continuous monitoring and early threat detection. However, the adaptation of network-level intrusion detection techniques to application-level contexts introduces challenges related to contextual interpretation and data granularity.

Despite the demonstrated benefits, several trade-offs must be considered. One of the primary concerns is the computational overhead introduced by multi-layered validation mechanisms. While early detection reduces long-term costs, the immediate impact on pipeline performance can be significant, particularly in large-scale systems. This aligns with the observations of Akkaya and Younis (2019), who highlight the need for optimization in resource-constrained environments. Balancing security and performance remains a critical

challenge that requires adaptive and context-aware validation strategies.

Another limitation is the complexity of integrating diverse tools and technologies within CI/CD pipelines. As highlighted in Cingil (2001) and Fox et al. (2000), system integration in distributed environments requires careful coordination and standardization. In practice, inconsistencies in tool compatibility and configuration can hinder the effective implementation of proactive validation frameworks. This challenge underscores the need for standardized protocols and modular architectures that facilitate seamless integration.

Furthermore, while the proposed framework demonstrates strong conceptual validity, its practical implementation may vary depending on organizational context, system architecture, and development practices. Factors such as team expertise, infrastructure capabilities, and organizational culture can influence the effectiveness of proactive validation. These contextual variables must be considered when generalizing the findings of this study.

In comparison with existing literature, this research provides a more holistic perspective by integrating workflow modeling, intrusion detection, and CI/CD practices into a unified framework. While prior studies have addressed these domains individually, the synthesis presented here highlights the interdependencies between them and demonstrates how their integration can enhance overall system security.

Overall, the discussion emphasizes that proactive safeguard validation is not merely a technical enhancement but a strategic approach to software development. Its successful implementation requires a balance between security, performance, and operational complexity, as well as a commitment to continuous improvement and adaptation.

6. Conclusion

This research has systematically examined the impact of proactive safeguard validation on early weakness identification within Continuous Integration and Continuous Delivery workflows. By synthesizing theoretical foundations from workflow modeling, intrusion detection systems, and distributed architectures, the study developed a comprehensive multi-layered validation framework designed to enhance early-stage vulnerability detection.

The findings demonstrate that integrating validation mechanisms at the initial phases of the development lifecycle significantly improves detection accuracy, reduces false positives, and minimizes the propagation of vulnerabilities into production environments. The incorporation of static, dynamic, and behavioral validation layers ensures a comprehensive assessment of system security, addressing multiple dimensions of risk. Furthermore, the alignment with empirical evidence from Thanvi et al. (2026) reinforces the effectiveness of shift-left security practices in modern software engineering.

A key contribution of this research lies in its holistic approach, which bridges the gap between theoretical models and practical implementation. By integrating intrusion detection methodologies into CI/CD pipelines, the study extends traditional security practices and provides a scalable solution for continuous validation. Additionally, the application of workflow modeling principles ensures that validation processes are seamlessly embedded within development pipelines, enhancing both security and operational efficiency.

However, the study also acknowledges several limitations, including computational overhead, toolchain complexity, and contextual variability. Addressing these challenges requires the development of adaptive validation strategies that balance security requirements with performance constraints. Future research should explore the integration of artificial intelligence and machine learning techniques to further enhance validation efficiency and adaptability.

In conclusion, proactive safeguard validation represents a critical advancement in secure software development. As CI/CD practices continue to evolve, the integration of early-stage security testing will become increasingly essential. This research provides a foundational framework for achieving this integration and highlights the broader implications of proactive validation for the future of DevSecOps.

References

1. Aalst, W. (2000). “Loosely coupled interorganizational workflows: modeling and analyzing workflows crossing organizational boundaries,” Information & Management, Vol. 37, pp. 67–75.
2. Akkaya, K., & Younis, M. O (2019). A survey of routing protocols in wireless sensor networks. Ad

Hoc Network Journal, 3 (3), 325 - 349.

3. Anjum, F., Subhadrabandhu, D., Sarkar, S., & Shetty, R. (2019). Optimal placement of intrusion detection modules in sensor networks. In Proceedings of the First International Conference on Broadband Networks (BROADNETS) (pp. 1 - 5).
4. Brownfield, M. (2019). Wireless sensor network denial of sleep attack. In Proceedings of the IEEE Workshop on Information Assurance and Security (pp. 1 - 6). United States Military Academy, West Point, NY.
5. Cingil, I. (2001). "An Architecture for Supply Chain Integration and Automation on the Internet," Distributed and Parallel Databases, Vol. 10 pp. 59–102.
6. Fox, M., Barbuceanu, M., and Teigen, R. (2000). "Agent-oriented Supply-chain Management," The International Journal of Flexible Manufacturing Systems, Vol. 12, pp. 165–188.
7. Hess, T., Rees, L., and Rakes, T. (2000). "Using Autonomous Software Agents to Create the Next Generation of Decision Support Systems," Decision Sciences, Vol. 31, No. 1, pp. 1–31.
8. Li, G., He, J., & Fu, Y. (2019). Group-based intrusion detection system in wireless sensor networks. Computer Communications, 31 (18), December 2019.
9. Liu, J., Zhang, S., and Cao, J. (2001). "An Inter-Enterprise Workflow Management System for B2B E-Commerce and Supply Chain," IEEE International Conference on Systems, Man, and Cybernetics, Vol. 5, pp. 2921–2926.
10. Sadotra, P., Sharma, C., & et al (2019). Intelligent intrusion detection system in computer security. International Journal of Computer Science and Mobile Computing, 5 (9), 23 - 28.
11. Y. S. Thanvi, K. Pappu and A. Parashar, "Effect of Shift-Left Security Testing on Early Vulnerability Detection in CI/CD Pipelines," SoutheastCon 2026, Huntsville, AL, USA, 2026, pp. 1-7, doi: 10.1109/SoutheastCon63549.2026.11476382.