Evolver

AI-Powered Sec/Dev/Ops

The Evolution of Secure Delivery

January 2026

Executive Summary

The traditional trade-off between software delivery speed and rigorous security is becoming obsolete. Faced with accelerating market demands and an increasingly sophisticated cyber threat landscape, organizations can no longer afford the “bolted-on” approach where cybersecurity is a final hurdle before deployment. Modern guidance such as NIST’s Secure Software Development Framework (SSDF) explicitly emphasizes integrating cybersecurity practices throughout the lifecycle and “shifting left” to address cyber vulnerabilities earlier in development. [1]

The future lies in a paradigm shift to Security/Development/Operations (Sec/Dev/Ops) powered by Artificial Intelligence (AI). This approach integrates AI-driven coding assistants, accelerated software testing, and low-code/no-code (LCNC) platforms within a framework of Cybersecurity by Design, where data security requirements are defined before software development begins. By shifting cybersecurity to the far left of the development lifecycle, organizations establish it as a foundational prerequisite for architecture, code, and data flows rather than an intermediary checkpoint. As a result, teams can significantly reduce rework, time, and cost while improving data integrity and data security posture when combined with sound governance and automation. [1][2]

This whitepaper outlines the components, strategies, and necessary mindset shifts required to implement AI-powered Sec/Dev/Ops in line with current federal and industry best practices. [1][2][3]

Cybersecurity Comes First: Defining the New Standard

The pressure to deliver digital capabilities faster has never been greater. However, the velocity of modern software development often outpaces the capacity of traditional cybersecurity measures, creating unacceptable organizational risk. The traditional industry response for software development has been the evolution of Development and Operations (DevOps) into Development, Security, and Operations (DevSecOps), integrating security practices into agile workflows. [2][3] We argue that this evolution does not go far enough. To truly meet the increasing cybersecurity demands of the modern cyber threat environment, organizations must adopt AI-powered Security, Development, and Operations (Sec/Dev/Ops).

Why the deliberate reordering of the acronym to put “Sec” first? We believe this step is necessary to focus on the urgent need for enhanced cybersecurity in software development. In the traditional “DevSecOps” model, security is often viewed as a gate or a mid-process integration point. In a “Sec/Dev/Ops” mindset, security is the foundation upon which software development and operations are built. [1][2] It is the prerequisite for writing the first line of code.

AI-powered Sec/Dev/Ops is the holistic fusion of advanced technologies and proactive methodologies. It means utilizing AI-powered coding assistants to write more secure code faster; leveraging low-code/no code (LCNC) platforms such as Appian, ServiceNow, and Salesforce to accelerate delivery while managing their unique security surfaces; and applying AI-powered software testing tools to predict and identify vulnerabilities earlier than humanly possible.[2][7]

Crucially, these technologies must be underpinned by a Cybersecurity by Design approach beginning with a focus on the integrity of the Software Bill of Materials (SBOM) and verifying data integrity throughout the lifecycle. [4][5] The net result is a system where speed and security are not competing forces, but symbiotic outcomes.

The Paradigm Shift: "Shift Left" and "Built-In" Security

To understand the power of AI-powered Sec/Dev/Ops, we must contrast it with legacy approaches. Historically, software was developed thrown "over the wall" to Quality Assurance (QA) testers, and finally subjected to a security review just prior to production. This bolted-on approach to security is inherently inefficient. Discovering a critical architectural flaw late in the software development life cycle (SDLC) requires expensive rework, delays time-to-value, and often leads to rushed patches that introduce new instabilities. [1] [3]

AI-powered Sec/Dev/Ops is predicated on the Shift Left paradigm. This philosophy mandates that integration of critical capabilities (most critically, cybersecurity aspects) occurs at the earliest possible stages of the software development lifecycle. Federal guidance such as NIST SP 800-218 and emerging DevSecOps practice guides explicitly highlight shifting security earlier in the lifecycle to reduce vulnerabilities and remediation costs. [1][2][3]

Cybersecurity must be built-in, not bolted-on. When security controls, automated vulnerability scanning, and compliance guardrails are embedded directly into Integrated Development Environments (IDEs) and Continuous Integration/Continuous Deployment (CI/CD) pipelines, software developers receive immediate feedback. They can address security flaws in real-time as they write code, rather than weeks later. This automated, built-in validation ensures greater efficiency, lower costs for intended outcomes, and improved software quality. It should be noted that runtime / right-side controls (monitoring, behavior analytics, incident response) remain essential; built-in controls complement, rather than replace, them. [2][3]

Cybersecurity by Design: The Foundation

Cybersecurity by Design is the methodology that makes "Shift Left" possible. It requires moving beyond reactive scanning and into proactive architecture. Before software development begins, cyber threat modeling must be conducted to identify potential attack vectors based on the system's intended design, data flows, and external dependencies. This analysis must explicitly account for development frameworks, open-source or public-domain components, and product-based developer tools that are incorporated into the solution, as each introduces distinct trust boundaries and risk considerations. [1][2]

A critical component of this approach is ensuring data integrity. In an AI-driven environment, the data used to train large language models (LLMs) or feed applications is as critical as the code itself. Cybersecurity by Design implements controls on data lineage, ensuring that the data have not been tampered with or poisoned, maintaining trust in the system outputs. [1][2][7]

The AI Accelerator: Enhancing Development and Testing

Once the “Sec-first” foundation is established, AI technologies act as powerful accelerators for the “Dev” and “Ops” components. Recent NIST and DoD work indicates that combining DevSecOps with AI and automation can improve efficiency and yield higher-quality software more quickly. [2][3][7]

AI-Powered Tools for Software Testing

Traditional software testing is often a bottleneck, relying on manual script creation and labor-intensive regression testing. AI transforms testing from a reactive validation step into a more predictive assurance activity. [2][7][8]

AI-powered testing tools can:

- Generative Test Creation: Automatically analyzes code and user stories to generate comprehensive test cases, ensuring higher code coverage rather than manual efforts alone.
- Self-Healing Tests: Automatically update test scripts when the underlying application changes, reducing the maintenance overhead of automated tests.
- Predictive Analysis: Utilizes historical data to predict where defects are most likely to occur, allowing software testing teams to focus testing efforts on high-risk areas.
- By automating complex software testing scenarios, AI ensures that security and functionality validate continuously at the accelerated speed of AI-powered software development. [2][7]

AI-Powered Tools for Software Development

AI-assisted coding tools have matured from simple autocomplete functions into sophisticated partners in software development. Integrated directly into the developer’s environment, these tools analyze context and leverage natural language processing (NLP) to suggest entire code blocks, functions, and documentation.

From a cybersecurity perspective, advanced AI coding assistants can be trained on vast datasets of both secure and insecure code patterns. They can proactively suggest secure coding practices in real time, flagging potential cyber vulnerabilities like SQL injection or cross-site scripting as the developer types code. This acts as an always-on security tutor, raising the baseline security capability of every developer on the team and reducing the burden on downstream security reviews. [7][8]

The Low-Code/No-Code (LCNC) Revolution

An integral part of modern Sec/Dev/Ops strategies involves leveraging Low-code/No-code (LCNC) platforms such as Appian, ServiceNow, and Salesforce. These platforms democratize development and drastically accelerate delivery by abstracting much of the underlying code. However, incorporating LCNC introduces unique cybersecurity considerations that must be addressed to maintain a Sec/Dev/Ops posture. [2][3]

Addressing Cybersecurity in LCNC Environments

There is a common misconception that LCNC platforms automatically handle all cybersecurity issues. While it is true that leveraging Software-as-a-Service (SaaS) and Platform-as-a-Service (PaaS) solutions within specialized infrastructure (such as Government/Military/Secret cloud environments) allows the organization to inherit strong infrastructure security controls from the vendor, this is only part of the equation. Security in LCNC operates on a shared responsibility model.

While the vendor secures the platform "kernel," the organization is responsible for securing what they build on top of it. The critical security challenges in LCNC lie in:

- **Data Interfaces and APIs:** LCNC applications rarely exist in a vacuum; they must integrate with existing enterprise systems. Securing the Application Programmable Interface (APIs) and connectors that facilitate these data flows is paramount to prevent unauthorized access or data leakage across system boundaries.
- **Business Logic Flaws:** It is very easy to rapidly build software applications with flawed permission structures or insecure workflows in visual designers. "Sec-first" means implementing rigorous governance guardrails within the platform to ensure software developers cannot accidentally expose sensitive data. The approach requires explicit definition and enforcement of user personas, roles, and mission functions, aligned to Zero Trust principles such as least privilege, continuous authorization, and explicit trust validation.
- **Configuration Management:** Misconfigurations of the software platform itself remain a leading cause of cloud security failures. [2][3]

An effective Sec/Dev/Ops approach applies the same rigorous cybersecurity testing, access controls, and API security standards to LCNC configurations as to traditional high-code software development.

Securing the Supply Chain: The Importance of SBOM

Modern software development is rarely written from scratch; it is assembled. Software developers rely heavily on open-source libraries, frameworks, and third-party components to build software applications quickly. This creates a complex software supply chain where a single cyber vulnerability deep in a dependency tree (e.g., the Log4j incident) can have catastrophic, widespread consequences. [6][7]

Cybersecurity by Design necessitates rigorous validation of this supply chain, as achieved through the Software Bill of Materials (SBOM). An SBOM is a formal, machine-readable inventory of software components and dependencies, information about those components, and their hierarchical relationships. It is the "list of ingredients" for any piece of software. In an AI-powered Sec/Dev/Ops environment, SBOM management is automated. [4][5]

Tools must be integrated into the CI/CD pipeline to automatically generate SBOMs with each releasable artifact, cross-reference components against known vulnerability databases called the Common Vulnerabilities & Exposures (CVEs) and assess the reputation and health of the open-source projects being utilized. [4][5][7] This allows organizations to quickly identify where newly discovered cyber threat vulnerability exist across their software application portfolio, turning a weeks-long discovery nightmare into a minutes-long database query.

AI-Powered Sec/Dev/Ops Implementation: Challenges & Best Practices

Transitioning to an AI-powered Sec/Dev/Ops model is a significant undertaking that involves cultural changes as well as technical shifts. Federal and industry frameworks consistently stress that success depends on integrating people, process, and technology over time. [1][2][3]

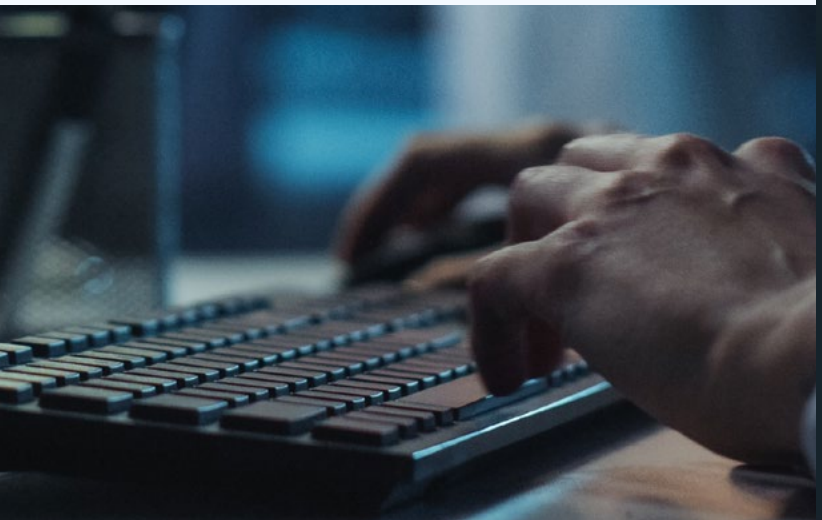
Challenges

Transitioning to an AI-powered Sec/Dev/Ops model is a significant undertaking that involves cultural changes as well as technical shifts. Federal and industry frameworks consistently stress that success depends on integrating people, process, and technology over time. [1][2][3]

- **Cultural Resistance:** Moving cybersecurity to the beginning of the process requires tearing down traditional silos between security, development, and operations.
- **AI Trust and Verification:** AI tools are probabilistic. Software developers must be trained not to blindly trust AI-generated code or test results, maintaining a “trust but verify” mindset to guard against AI hallucinations or subtle errors. [7][8]
- **Toolchain Complexity:** Integrating AI assistants, LCNC platforms, security scanners, SBOM generators, and pipeline orchestration tools requires careful architectural planning to avoid unmanageable complexity.

Best Practices

- **Define Governance First:** Establish clear policy guardrails for AI usage and LCNC development before widespread deployment. [1][2]
- **Automate Everything:** Manual steps are the enemy of Sec/Dev/Ops. Cybersecurity policies, testing, and SBOM analysis must be codified and automated within repeatable, verifiable pipelines, minimizing manual steps in the critical path of the pipeline. [1][3]
- **Continuous Investment in People:** The tools are only as good as the operators. Invest in training software developers on secure coding practices and training security professionals on modern development workflows. [2][3][7]



Benefits of AI-Powered Sec/Dev/Ops

By successfully implementing this model, organizations realize compound benefits aligned with current secure software development guidance. [1][2][3]

1

Velocity with Confidence

The primary benefit is the ability to release software faster without compromising security. The “Shift Left” approach means security approvals happen continuously eliminating last-minute deployment blockers.

2

Reduced Cost and Effort

Fixing security defects during design or coding is exponentially cheaper than addressing them in production. AI automation further reduces the manual effort required for coding and testing. [1][2][3]

3

Enhanced Risk and Posture

Utilizing Cybersecurity by Design and rigorous supply chain validation (SBOMs) can significantly improve visibility into cybersecurity supply-chain risk and enable reduction of overall cyber-attack surface through targeted remediation. [4][5][7]

4

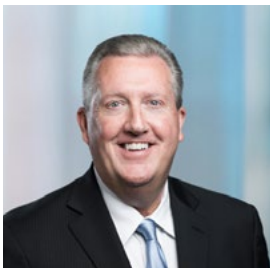
Greater Efficiency

“Built-in” security frees up high-value human resources. Cybersecurity teams spend less time on routine vulnerability management and more time on strategic cyber threat modeling, while software developers spend less time on rework.

Summary & Conclusion

The future of software delivery is not about choosing between speed and cybersecurity; it is about fusing them into a single, indistinguishable capability. AI-powered Sec/Dev/Ops represents the maturation of this concept. By adopting a “Sec-first” mindset, leveraging the proactive power of Cybersecurity by Design and utilizing AI to accelerate both traditional and low-code development, organizations can build resilient software factories. This approach ensures that as the pace of digital innovation accelerates, the integrity and cybersecurity of the systems we build keeps pace. [1][2][4][5]

About the Authors



Gregory A. Garrett is the Chief Operating Officer of Converged Security Solutions (CSS), leading expert teams within business units Evolver and eVigilant. These groups provide advanced IT, AI, cybersecurity, electronic security, and eDiscovery services to federal agencies and Fortune 500 companies.

With over 25 years of executive experience and a background as a CIO, CTO, CISO, and COO, he has managed more than $40 billion in global tech contracts and served in key roles across both large and mid-sized firms.

A retired U.S. Air Force Colonel and accomplished author, Garrett has also contributed extensively to industry thought leadership through 24 books, more than 150 articles, expert testimony, and over 250 speaking engagements.

Dr. Brian McElyea is an accomplished cybersecurity executive recognized for leading largescale digital transformation initiatives and building high-performing Cyber Centers of Excellence that drive mission-aligned security outcomes.

With over two decades of leadership across healthcare, defense, and federal sectors, he has guided global organizations through the complexities of risk management, compliance, and modernization with measurable results.

www.evolverinc.com

Works Cited

- **[1]** Souppaya, Murugiah, et al. Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities. *NIST Special Publication 800-218*, National Institute of Standards and Technology, Feb. 2022
- **[2]** National Cybersecurity Center of Excellence (NCCoE). DevSecOps—Communicating and Integrating Secure Software Development Practices. *NIST SP 1800-44 (Draft)*, National Institute of Standards and Technology, 2025.
- **[3]** Department of Defense Chief Information Officer. DoD Enterprise DevSecOps Fundamentals v2.5. 23 Oct. 2024.
- **[4]** U.S. Department of Commerce, National Telecommunications and Information Administration. The Minimum Elements for a Software Bill of Materials (SBOM). 12 July 2021.
- **[5]** Cybersecurity and Infrastructure Security Agency (CISA). 2025 Minimum Elements for a Software Bill of Materials (SBOM). 22 Aug. 2025.
- **[6]** Cybersecurity and Infrastructure Security Agency (CISA). Mitigating Log4Shell and Other Log4j-Related Vulnerabilities (AA21-356A). 23 Dec. 2021; and Review of the December 2021 Log4j Event, Cyber Safety Review Board, July 2022.
- **[7]** Cybersecurity and Infrastructure Security Agency (CISA). 2022 Top Routinely Exploited Vulnerabilities (AA23-215A). 3 Aug. 2023.
- **[8]** Continuum GRC. "What Role Does Cloud Automation and AI Play in NIST 800-218 Compliance?" 14 Aug. 2024, and Assyst. "The Future of DevSecOps, Testing and AI-Driven Software Delivery." Accessed 2025.

www.evolverinc.com