Building Secure Software at Speed: Four Emerging DevSecOps Trends for Government

How agencies are using automation to accelerate development, strengthen security, and modernize software operations.

Artificial intelligence is rapidly reshaping software development, enabling agencies to build applications faster and automate tasks that once required significant manual effort. However, organizations are discovering that increased speed must be balanced with stronger controls.

The next phase of DevSecOps automation focuses on ensuring that AI-generated code, open source components, and development decisions align with security, compliance, and software quality requirements from the outset.

Some organizations are automating software maintenance and remediation, while others are modernizing authorization processes, rethinking infrastructure architectures, or applying AI-driven guardrails to software development.

In this report, four industry experts share their perspectives on the technologies, strategies, and automation capabilities shaping the future of DevSecOps in government.

First-Party Software Support for Critical Infrastructure

AI is not only accelerating the pace of software development but also dramatically increasing the volume and speed at which vulnerabilities are created, discovered, and exploited. What once took weeks or months to identify and weaponize can now happen in days—or even hours.

At the same time, organizations can no longer afford to focus solely on patching the most severe flaws.

As John Gozzi, director of Tanzu public sector & healthcare solution architecture at Broadcom, explains, "You really need to be focused on all vulnerabilities because with AI-driven analysis, you can take a bunch of small, maybe unimportant vulnerabilities and turn them into a critical one."

This means that delayed patching of lower-priority vulnerabilities is increasingly risky.

Reducing Vulnerability Exposure Windows

For agencies that depend on mission-critical applications built on open source frameworks, libraries, containers, and runtime services, shrinking remediation windows are driving the adoption of continuously secured software delivery models. These approaches emphasize automated remediation and platform-managed operations to minimize exposure to vulnerabilities and accelerate risk reduction.

One emerging DevSecOps trend is first-party software support. Rather than relying solely on community-maintained open source releases or third-party support providers, organizations are turning directly to the original maintainers of critical software platforms and frameworks. This approach provides earlier access to security fixes, authoritative guidance, and faster remediation timelines.

For agencies running mission-critical Spring-based Java applications, first-party support can provide security updates before they are broadly available through the open source ecosystem. Earlier this year, community-reported vulnerabilities affecting Spring increased by 1,700%, underscoring how rapidly the security landscape is evolving with the emergence of frontier AI models.

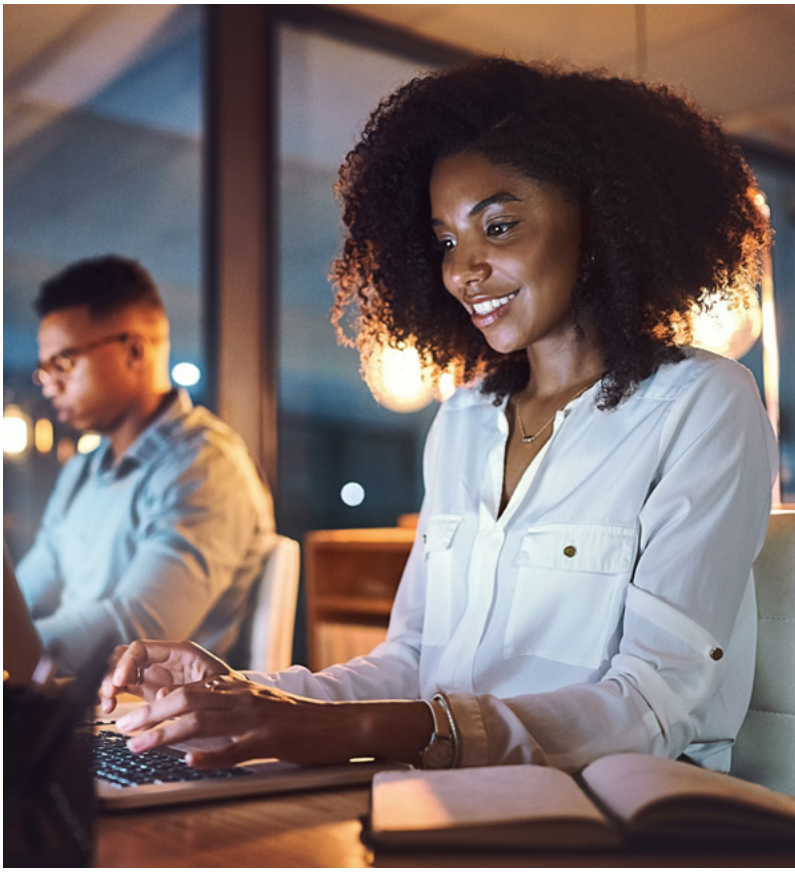
Gozzi notes that critical infrastructure organizations using first-party support can receive fixes up to two weeks earlier than open source releases, significantly reducing their exposure to newly discovered vulnerabilities.

Reducing Attack Surfaces

Comprehensive patch management remains essential, but organizations are increasingly recognizing that remediation alone cannot solve the problem.

The growing volume of vulnerabilities discovered across open source software is forcing agencies to focus not only on how quickly they patch, but also on reducing the number of vulnerabilities introduced into their environments in the first place.

As open source software becomes embedded throughout modern application portfolios, agencies are placing greater emphasis on trusted software supply chains, curated software artifacts, and continuously maintained open source components.



These approaches help reduce attack surface by minimizing exposure to known vulnerabilities before software is deployed into production environments.

This shift reflects a broader DevSecOps trend toward preventative security. Rather than relying solely on downstream vulnerability remediation, organizations are adopting software components that are continuously rebuilt, maintained, and validated against emerging threats.

By reducing the number of vulnerable components entering the environment, agencies can decrease operational risk while improving software supply chain integrity.

Vendor-maintained software components also provide operational advantages. Agencies gain access to hardened software artifacts, documented software bills of materials (SBOMs), and security-focused supply chain controls that simplify compliance efforts while reducing the burden of managing open source vulnerabilities internally.

Perhaps most importantly, continuously secured software delivery enables agencies to focus on mission outcomes rather than remediation cycles.

> "The goal is to reduce the burden of keeping software secure and up to date, so organizations can maintain continuity, lower risk, and move modernization efforts forward on realistic operational timelines."
>
> – John Gozzi, Broadcom

Decoupling Security and Networking From Applications and Cloud Infrastructure

As government agencies accelerate the deployment of cloud-native applications, APIs, and AI-enabled services, many are discovering that traditional approaches to security and networking are slowing innovation and increasing complexity.

According to Chris Holmes, CEO of Greymatter.io, most organizations still rely on "bolt-on security" approaches. Application teams often embed security controls, certificates, logging mechanisms, and access policies directly into their code.

At the same time, cloud administrators build security policies into the infrastructure itself using cloud-specific tools and services. The result is a fragmented security model that can be difficult to manage and even harder to replicate across environments.

This piecemeal approach becomes especially problematic when agencies need to move applications between cloud environments or deploy them into tactical or disconnected environments.

Security controls often must be rebuilt from scratch, increasing costs, delaying deployment timelines, and creating inconsistencies that complicate security audits and authorization efforts.

Creating a Portable Layer

To address these challenges, a growing number of government organizations are moving toward a DevSecOps model that decouples security and networking from both application code and cloud infrastructure.

Instead of embedding security into individual applications or cloud platforms, agencies are implementing portable policy enforcement layers that sit between applications and infrastructure.

Enforcement happens at each workload, where each gets its own identity, authenticated via mutual TLS. Additionally, policy is applied at the proxy enforcement point, so granular security controls travel with the workload rather than living in the application or the cloud platform.

> "This approach creates a consistent control layer that pulls control away from the application teams and decouples it from the infrastructure, so the policy is portable and moves with the workload instead of being rebuilt for every environment."
>
> – Chris Holmes, Greymatter.io

This trend is closely aligned with broader Federal priorities around Zero Trust and continuous Authorization to Operate (cATO). In practice, that assurance is delivered as evidence-as-code continuous monitoring. This is aligned with NIST 800-137, giving authorizing officials a live view of the control state rather than a point-in-time snapshot.

Less Time Spent Maintaining Security Controls

The benefits of adding a portable policy enforcement layer are significant. First, portability enables agencies to deploy applications and AI models across clouds, networks, and mission environments without having to rebuild security architectures each time.

For regulated and classified environments, the base enforcement layer itself is FIPS 140-3 validated encryption, so its cryptographic controls meet Federal requirements without additional integration work. This lets defense and intelligence customers enforce consistent encryption across application traffic, whether deployed to the cloud, on-prem, or tactical.

Second, standardized security controls simplify command, control, auditing, and compliance by providing a consistent security model across applications. In one recent case, a Department of War customer platform achieved 62% faster compliance audits with this approach.

Because security controls are defined as declarative policy-as-code and GitOps-managed, every change pull-request is reviewed, fully audited, and instantly reversible. Compliance evidence is continuously generated and reproducible, compressing audit timelines while strengthening governance.

Operational efficiency is another major advantage. Automating tasks such as certificate management, audit logging, and policy enforcement reduces the burden on both developers and infrastructure teams. Organizations can spend less time maintaining security controls and more time delivering mission capabilities.

Ultimately, decoupling security and networking from applications and infrastructure helps agencies achieve faster deployment cycles, stronger security consistency, lower operational costs, and a more agile path toward Zero Trust and continuous authorization.

"Processes that used to take months now take days, in some cases hours, because the security controls are repeatable and far easier to validate," says Holmes.

Modernizing the ATO Process

Federal agencies are under increasing pressure to adopt new cloud services, AI-enabled applications, and digital capabilities at mission speed. Yet many continue to rely on authorization processes built for a slower era of technology.

Traditional Authorization to Operate (ATO) and Risk Management Framework (RMF) reviews often depend on static documentation, screenshots, spreadsheets, and point-in-time assessments that can take months to complete. While these processes were designed to ensure security and compliance, they can also delay the delivery of critical capabilities to the people who need them most.

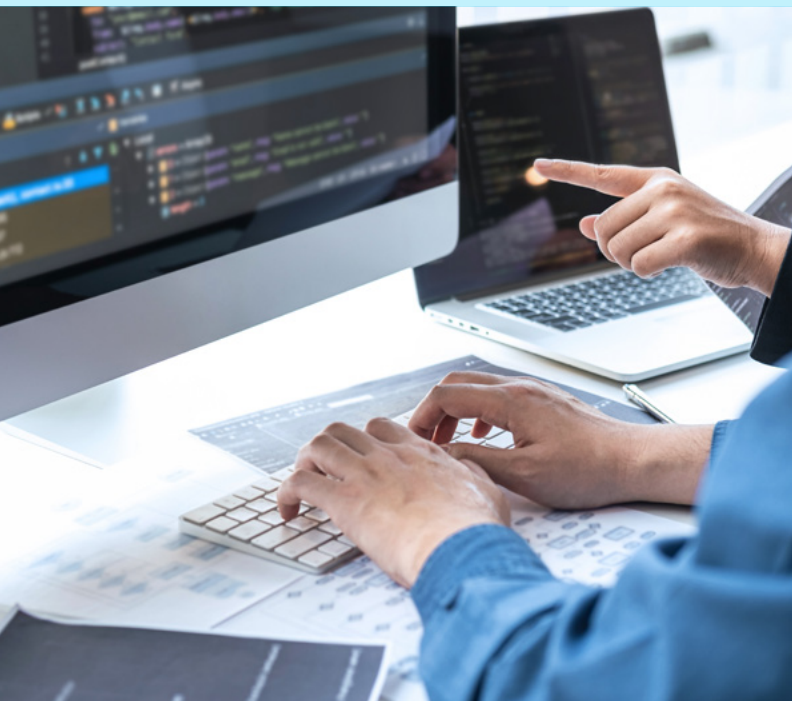
According to Josh Krueger, CISO/FSO at ProjectHosts, the challenge facing agencies today is not simply how to deploy software faster, but how to get trusted software into agency hands more quickly, more reliably, and with better evidence behind the risk decision.

Agencies need to modernize without sacrificing confidence in their security, compliance, and authorization posture. Krueger notes that many organizations remain "stuck with this long, arduous six-step RMF process" that does not align well with modern software development and cloud operations.

Traditional ATO: Not Keeping Up With Change

The problem is that today's technology environments are far more dynamic than the authorization processes used to evaluate them. Cloud-native architectures, AI-enabled services, software supply chains, and continuous software updates mean systems can change significantly after an initial assessment. A security review conducted months earlier provides only a snapshot in time.

> "A one-time-a-year assessment with monthly vulnerability scans doesn't really tell us a whole lot about the product. It says that at one time of the year it was secure, but there are no assurances following the assessment that things remain in a secure state."
>
> – Josh Krueger, ProjectHosts



Continuous Evaluation of Trustworthiness

Agencies and software providers are increasingly moving toward a model built on continuous evidence and persistent validation. This shift aligns with broader Federal initiatives, including FedRAMP 20X, continuous monitoring programs, software supply chain security efforts, and the growing use of AI-enabled software delivery.

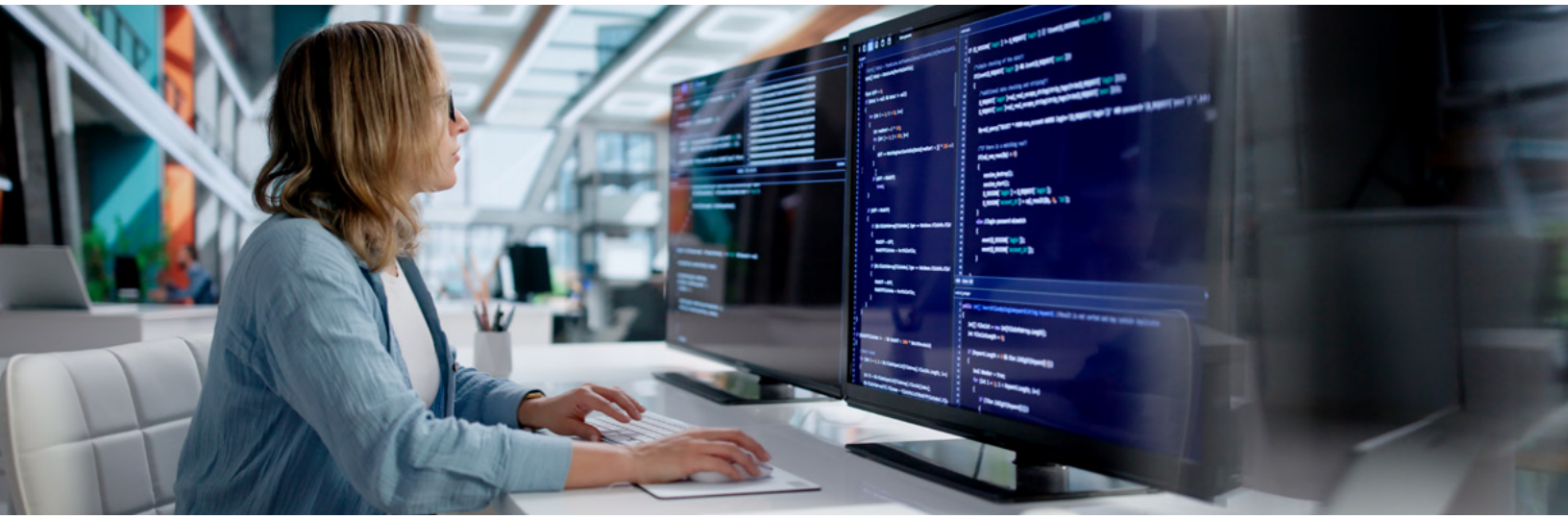
Rather than relying solely on static ATO packages and periodic assessments, organizations are seeking ongoing visibility into what has changed, what has been tested, which vulnerabilities exist, which controls are operating, and which risks remain.

This approach allows authorizing officials to continuously evaluate whether a system remains trustworthy after the initial authorization decision.

The goal is not to eliminate security reviews, but to strengthen them through better, fresher, and more actionable evidence. Instead of reviewing hundreds of pages of documentation and historical reports, agencies can leverage automated telemetry and real-time validation to understand a system's current security posture.

The benefits are significant. Continuous evidence and persistent validation improve the quality of risk decisions by providing visibility into actual operating conditions rather than historical snapshots.

Krueger explains, "Persistent validation speeds up the assessment and reporting timeline, so it cuts days, weeks, and even months from the overall ATO process."



Automating Consumption of Open Source Components

Agencies are seeing dramatic gains from adopting coding assistants such as GitHub Copilot, Cursor, and Claude Code to accelerate the development of new applications and refactor aging, monolithic backend systems.

However, Tom Tapley, Federal product manager at Sonatype, cautions that the same tools that boost productivity can also generate low-quality code, introduce hidden security vulnerabilities, and recommend unsafe open source components. The underlying issue is that they often lack current intelligence about software quality and security.

Tapley explains that AI models "don't have the best information about what versions to pick" and often cannot distinguish between trusted open source projects and malicious packages designed to mimic them.

The result is a new class of software supply chain risk. AI-generated code may pull in vulnerable dependencies, recommend outdated versions, or even hallucinate components that do not exist.

> "Sonatype's 2026 State of the Software Supply Chain Report found that AI coding assistants recommended non-existent package versions nearly 28% of the time—creating manual rework and delays when applications fail to build."
>
> – Tom Tapley, Sonatype

More concerning, some recommendations can introduce components that enable remote code execution or other forms of compromise. "AI will do that unwittingly," Tapley says, because it lacks real-time awareness of emerging vulnerabilities and malware.

MCP Servers as Intelligent Filtering Layer

To address the problem, organizations are turning to DevSecOps tools for automated screening and governance of open source components.

A key trend is the use of Model Context Protocol (MCP) servers as an intelligent filtering layer between coding assistants and software repositories.

Tapley describes MCP servers as a "guardrail" that evaluates component requests against current vulnerability data, malware intelligence, and organizational policies before they are approved for use.

If an AI tool selects a vulnerable or malicious package, the MCP server intercepts the recommendation and steers the assistant toward a safer, policy-compliant version before it reaches an artifact repository.

Automation is also transforming remediation. Rather than waiting for vulnerabilities to be discovered later in the development lifecycle, AI-enabled systems can recommend compliant versions, verify compatibility, and even update code automatically.

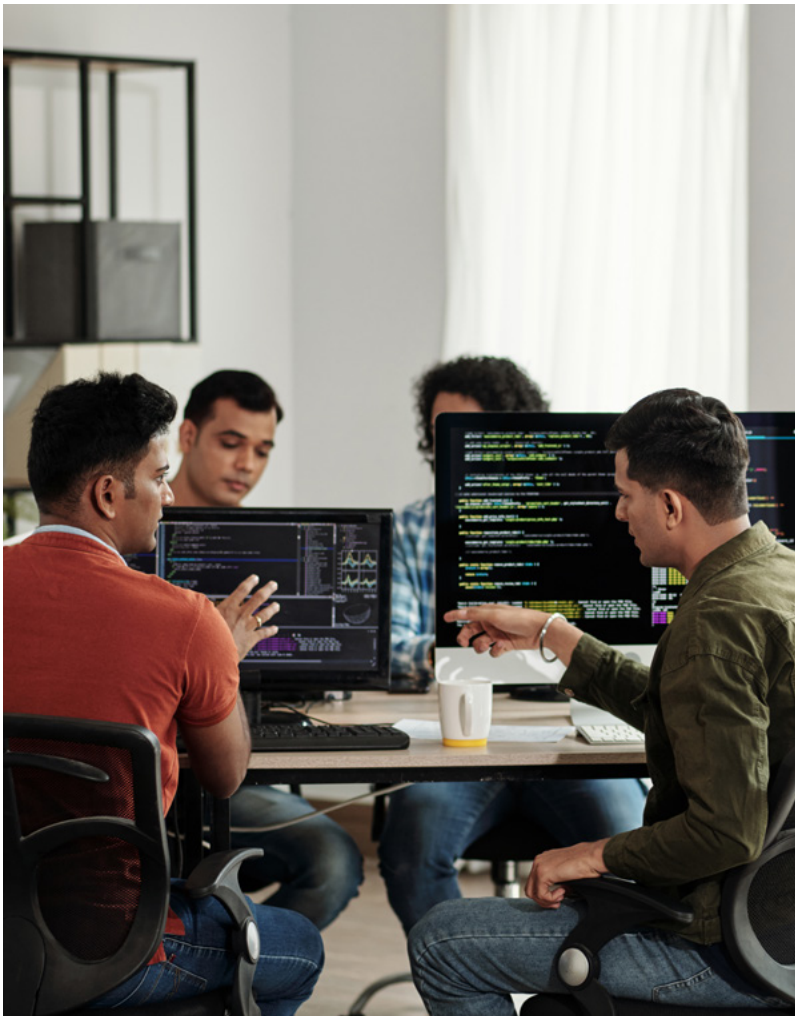
Address Issues Ahead of Production

Tapley predicts that AI-driven guardrails will enable organizations to address many software issues before they ever reach production. "A lot of remediation effort is going to be eliminated by these preventive measures," he says.

AI is also helping automate waiver management by generating documentation, identifying policy-compliant alternatives, and accelerating approval workflows for exceptions that cannot be resolved immediately.

The benefits are substantial: fewer vulnerable or malicious components entering applications, reduced exposure to software supply-chain threats, less time spent investigating AI hallucinations, faster remediation cycles, and significantly lower manual effort.

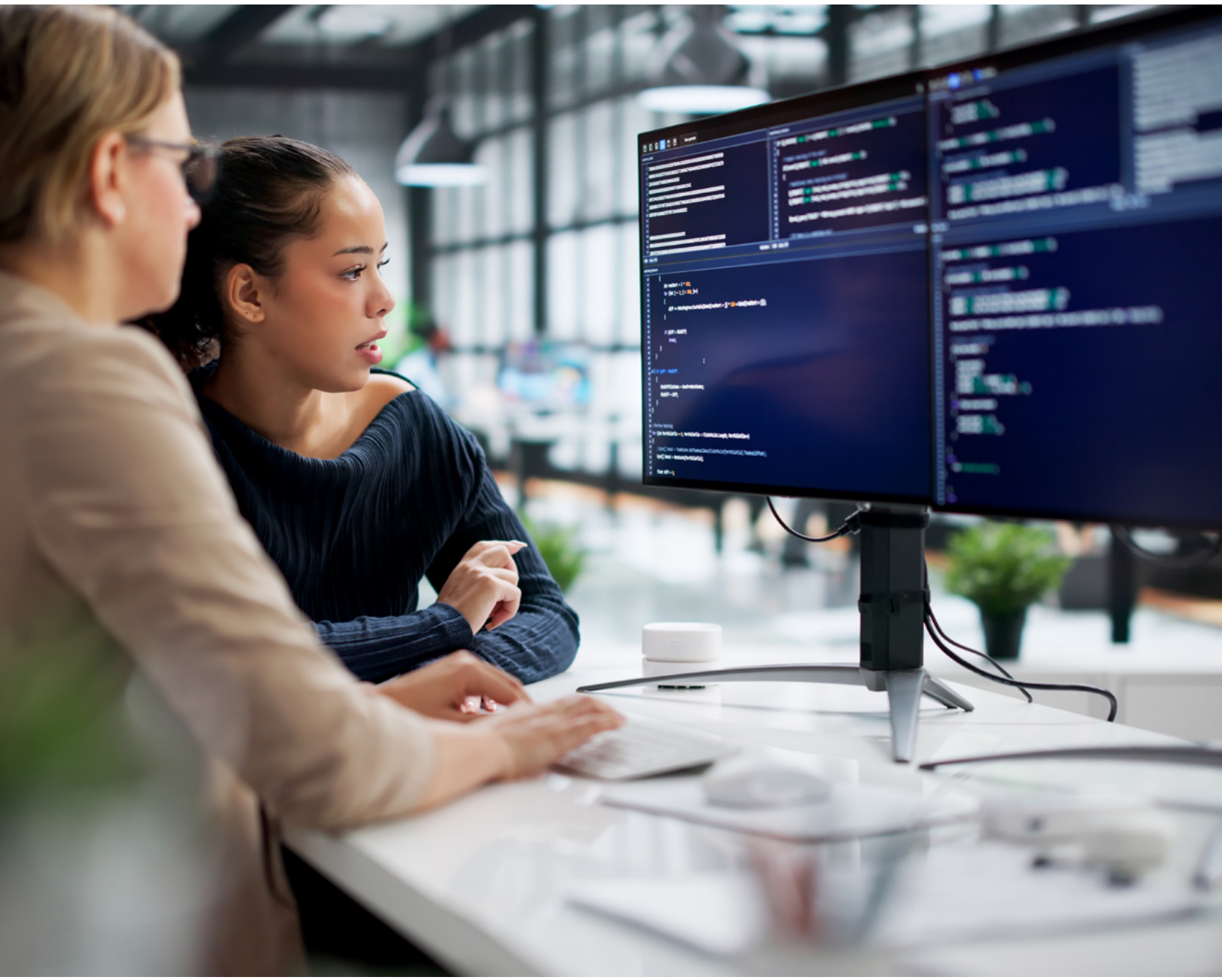
As Tapley explains, the goal is to "build clean, non-vulnerable code from the start, then remediate code in a very automated way."



Looking Ahead

As agencies expand their use of AI-assisted development, DevSecOps practices are evolving from detecting problems after the fact to preventing them in real time. Automated guardrails, policy enforcement, and AI-driven remediation are helping organizations reduce risk while maintaining development velocity.

The result is a more proactive approach to software delivery where teams move faster without sacrificing security, compliance, or code quality.



Carahsoft partners with leading software development companies to deliver innovative DevSecOps tools to Public Sector agencies, ensuring compliance by building security into every phase of the DevOps lifecycle. These cutting-edge solutions facilitate continuous Authority to Operate (cATO) initiatives by enabling collaborative planning, agile code optimization, flexible infrastructure management, automated system testing, and vulnerability scanning.

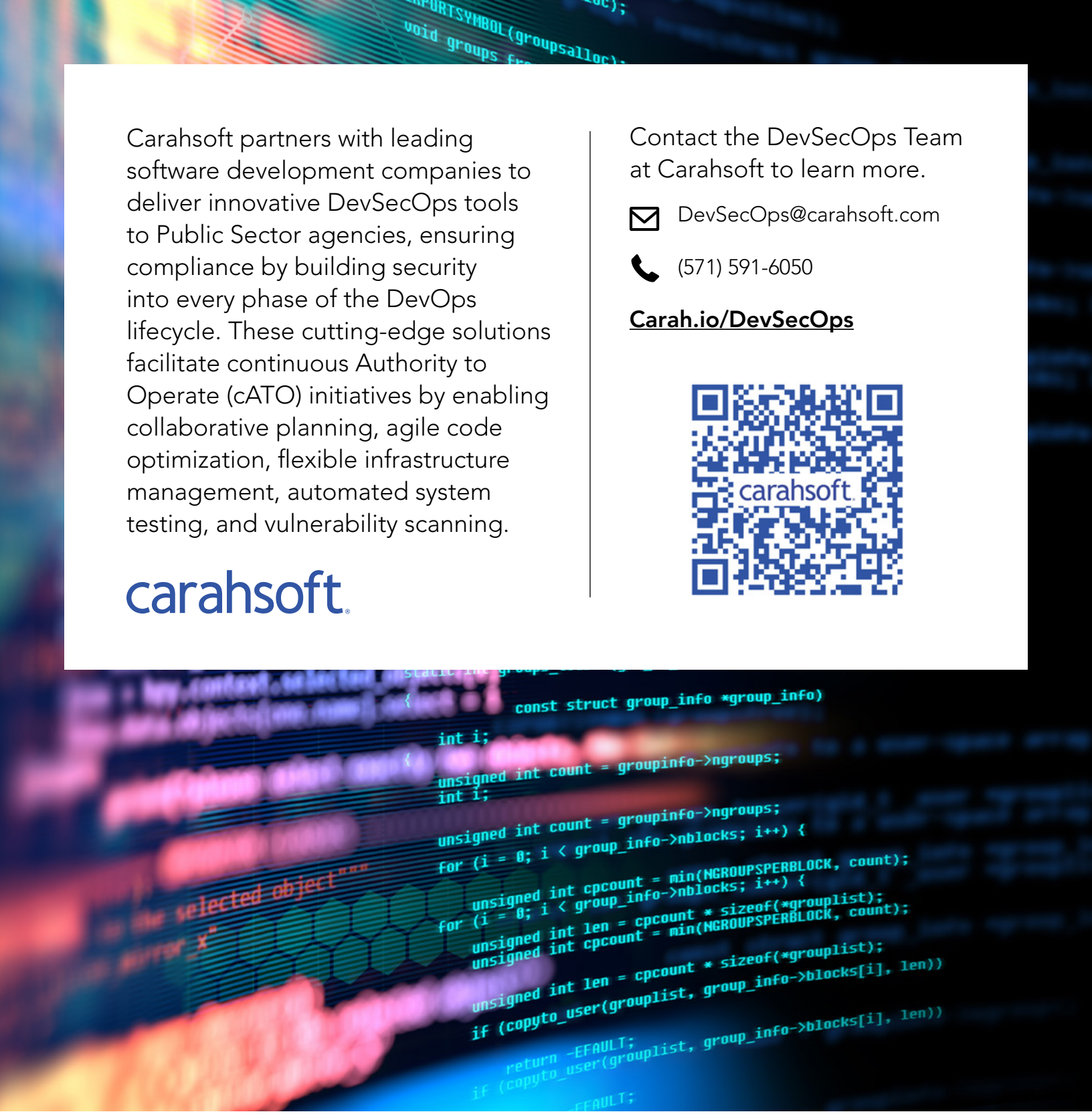
carahsoft.

Contact the DevSecOps Team at Carahsoft to learn more.

DevSecOps@carahsoft.com

(571) 591-6050

Carah.io/DevSecOps



GovWhitePapers

For additional resources on DevSecOps, visit GovWhitePapers.com