perforce

Build Security into the Code

A Security Leader's Governance Framework for Preventing Vulnerabilities at the Source

Cut through the noise and reduce false positives with security-focused static analysis (SAST) that works. This framework helps you build a static analysis program that engineering teams actually use, and that produces board-ready evidence of reduced software risk.

The Business Impact of Catching Defects Late

85%

Defects introduced during coding stage (Capers Jones)

640x

Potential cost at release if defects are not found (Capers Jones)

$10.22M

Average U.S. data breach cost (IBM 2025)

When a security defect is caught during the development phase, it is just a bug. When it escapes into production, it becomes a public vulnerability requiring customer notification, supply chain risk assessment, and board disclosure.

5 Governance Decisions That Enable a Streamlined, Developer-First Security Program

Use this framework to evaluate, restart, or fix your static analysis program. Each decision maps to a measurable outcome your board, auditors, and insurers can verify.

1. **Define What You're Solving For:**
   SAST tools can check thousands of rules, whether for established security standards or custom guidelines. While this provides a wide range of insights, analysis results can be noisy if not focused on the right issues. Most first attempts to successfully introduce SAST fail because they turn everything on. Narrow your scope: identify security defects (buffer overflows, injection, null pointer dereferences) that become vulnerabilities. Turn off style rules, naming conventions, and nice-to-haves. Focus only on findings that represent actual risk to your customers and your business.
2. **Agree on Meaningful Quality Gates With Engineering:**
   Security and development must jointly define which vulnerability classes are unacceptable in production. Document these as enforceable gates in your CI/CD pipeline to help ensure that enforcement is focused on material risk, rather than treating all findings equally. This is a collaborative agreement, not a security mandate. When both teams own the standard, compliance follows naturally and velocity increases.
3. **Make Engineering the Owner, Security the Enabler:**
   The security leader's role is to govern, monitor, measure — and also to educate. Enabling engineering teams to triage results frees up security teams to focus more on the most critical risks. Empower engineering teams by embedding analysis feedback directly into developers' IDEs, with clear, context-aware remediation guidance. When security frontloads effort in education, tooling integration, and telemetry setup, the business runs without daily security involvement, empowering engineers and freeing security to focus on exceptions.
4. **Measure What the Board Needs to See:**
   Track metrics that translate to business risk, not just vulnerability counts. Recommended KPIs: security defect escape rate (vulnerabilities reaching production vs. caught in development), mean time to remediation by severity, compliance coverage percentage against required standards and guidelines (CERT, CWE, OWASP), and cost-per-defect at each stage of the pipeline.
5. **Generate Compliance Evidence Automatically:**
   Your SAST tool should produce audit-ready reports that satisfy multiple frameworks simultaneously: NIST cybersecurity and CIS Control 18, ISO 26262 for automotive, IEC 62443 for industrial, IEC 62304 for medical devices, and PCI DSS for payment software compliance. One investment in properly certified static analysis satisfies requirements across all of these, plus supports cyber insurance underwriting documentation.

Why Using SAST Can't Wait: Emerging Risk

40%
Defects In AI-Generated Code

AI-Generated Code Is Insecure at Scale. Approximately 85% of developers are using (or planning to use) AI coding tools (Stack Overflow 2025), but code quality is decreasing with the use of AI-generated code. In a recent study completed, it was discovered that 40% of AI-authored outputs contained security vulnerabilities (IEEE, 2025). SAST is the only automated control that catches these defects before they enter production.

82%
Denied

Cyber Insurers Are Mandating Security Controls. 82% of denied cyber insurance claims involved missing controls (Coalition 2024). Insurers require NIST CSF or ISO 27001 alignment, which include the use of a static analysis tool. Certified static analysis provides documented proof of proactive vulnerability management during underwriting.

Why Security Leaders Choose Perforce Static Analysis Tools QAC And Klocwork

DevSecOps Ready	QAC and Klocwork are designed to work with existing CI/CD pipelines and are easy to automate. Perforce Static Analysis tools provide differential analysis, containerized builds, project streams, and risk prioritization capabilities that make it easy for developers to safeguard software from vulnerabilities with every commit. Intuitive guardrails for AI-generated code and AI-assisted code remediation with contextual fix suggestions for faster implementation.
Governance Built In	Validate platform delivers centralized compliance reporting, deviation workflows, trend analytics, and multi-project dashboards. Generates compliance reports without manual effort. Audit-ready evidence for your board, insurers, and regulators.
Precision and Speed	QAC offers configurable analysis depth and exhaustive abstract interpretation via inter-procedural dataflow analysis at speeds never before seen. Recommended rulesets target security defects, not code style. Accurate diagnostics with results developers trust — while maintaining development velocity. Klocwork SAST scales to scan very large, complex, multi-language codebases with millions of lines of code and identifies security issues efficiently for mission-critical projects of any size.
IDE-First Workflow	QAC and Klocwork integrate into Visual Studio, VS Code, Eclipse, and more. AI-assisted code remediation feature works with any AI code assist agent supporting MCP, including GitHub Copilot, Google Code Assist, and Amazon Q Developer. CI/CD integration via Jenkins, REST API, and containerized builds.
TÜV SÜD Certified	Independent third-party certification for ISO 26262, IEC 61508, IEC 62304, EN 50716. No general-purpose SAST competitor holds equivalent safety certifications.
Multi-Standard and Programming Language Coverage	Covers CWE Top 25, OWASP Top 10, CERT, MISRA, DISA STIG, PCI DSS. Accelerates compliance to new AI-specific standards and governance such as NIST SP 800-218A and OWASP Top 10 for LLM applications. Supports C, C++, C#, Java, JavaScript, Python, Kotlin, Rust.

Catch Security Defects Before They Become Threats: Build Secure Software with Static Analysis

Sign up for a custom demo to see how Perforce Static Analysis can work for your next project.

www.perforce.com/products/sca/custom-demo

Sources: IBM Cost of a Data Breach 2025; Jones, Capers: Applied Software Measurement: Global Analysis of Productivity and Quality; Stack Overflow Developer Survey 2025; Controneo, Domenico, Improta, and Liguori, "Human-written vs.ai-generated code: A large-scale study of defects, vulnerabilities, and complexity," 2025 IEE 36th International Symposium on Software Reliability Engineering (ISSRE), IEEE, 2025; Coalition Claims Report 2024.