opentext™

GUIDE

AI AppSec, proven at scale: OpenText's blueprint and outcomes

Practical metrics and lessons from an enterprise deployment that sped secure delivery

Contents

Executive summary	3
The challenge: Scaling AppSec across thousands of applications	4
The solution: Application Security Aviator	4
Results at scale	5
Broader context: OpenText, AI, and the future of AppSec	6
What’s next: Autonomous remediation and beyond	6
Conclusion	7

"At our scale, reviewing every finding manually was unsustainable. Aviator gave us a way to maintain depth of coverage without compromising velocity."

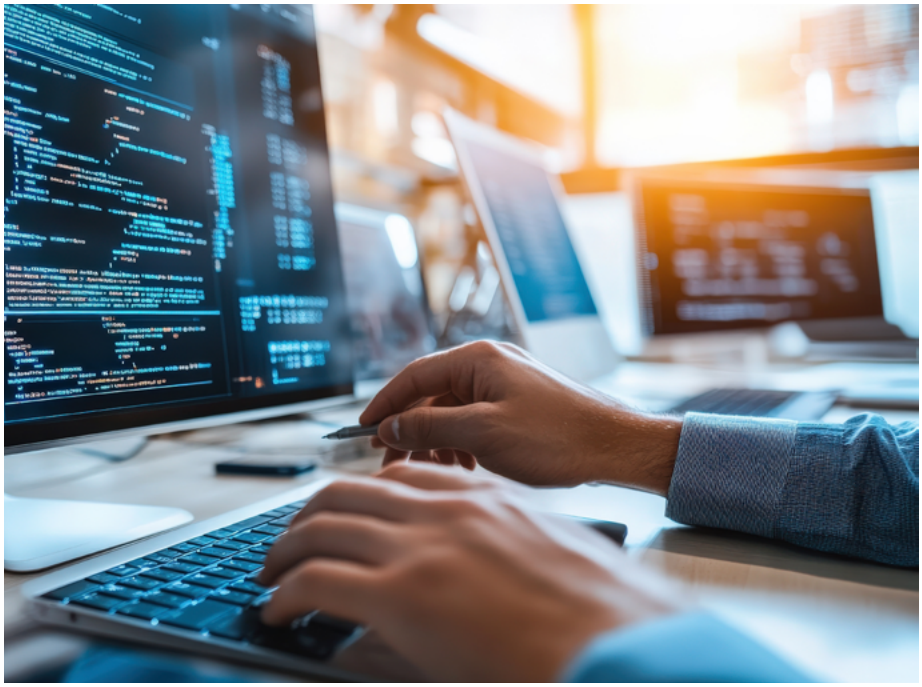
Dan Nolan,
Director, Software Engineering Research & Development, OpenText

Executive summary

OpenText is one of the world's largest software companies, developing and maintaining thousands of enterprise applications across a vast and complex ecosystem. With more than **7,000 developers** and an extensive technology footprint, ensuring software assurance at this scale requires more than traditional application security methods. It demands a proven, enterprise-grade solution. That's why OpenText implemented its own Application Security portfolio across the organization. Our tools proved to be the best fit to manage the complexity of our development environment, scalable, intelligent, and deeply integrated into our CI/CD processes.

But as the pace of innovation accelerated, so did the need for speed and precision in handling an ever-growing volume of security findings. This challenge led to the creation of OpenText™ Application Security Aviator™, our very own generative AI innovation built to make AppSec faster, smarter, and more accurate. It helps our teams prioritize risk, eliminate false positives, and stay ahead of emerging threats.

Within its first eight weeks, Aviator audited more than **300,000 findings across 1,500 applications**, reducing mean time to triage by **70 percent** and saving an estimated **three million minutes of manual review time**.



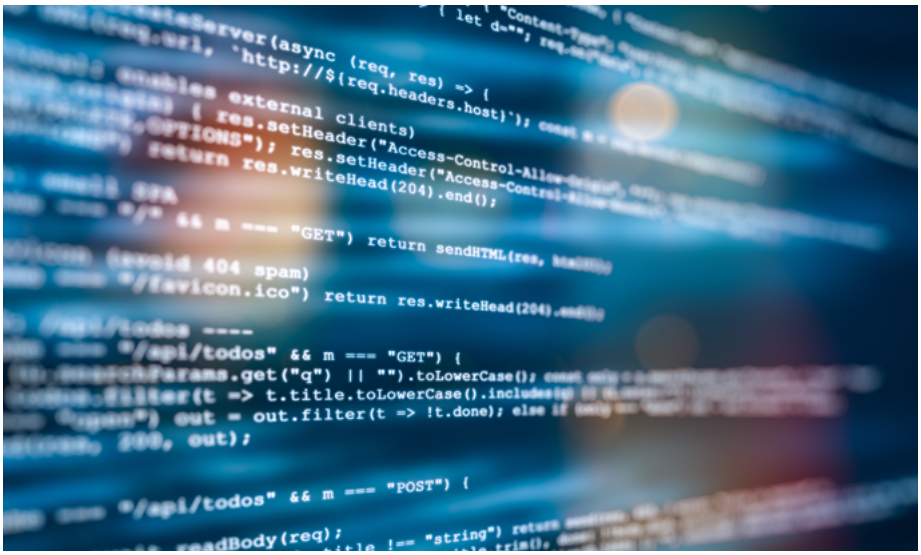
> **"Security teams can't win by doing more manual work. They win by automating intelligently—and Aviator is exactly that."**
>
> Frans van Buul,
> Director, Product Management
> Research & Development,
> OpenText

The challenge: Scaling AppSec across thousands of applications

Application security has evolved into a core enterprise function, but one persistent challenge remains: overwhelming volume. Static and dynamic testing generate essential insights, but at scale, they also produce tens of thousands of findings—each requiring validation, triage, and context.

At OpenText, the scale was substantial. With thousands of applications under management, security scans produced hundreds of thousands of issues, the majority of which needed manual review. On average, each finding required 10 minutes to assess, creating a massive operational burden that diverted engineering hours from product delivery to verification.

This was not a gap in coverage—it was a gap in scalability. The challenge isn't about finding security issues; it's about efficiently triaging, prioritizing, and acting on them at enterprise scale.



The solution: Application Security Aviator

Application Security Aviator is a generative AI capability within the OpenText Application Security platform. Built to accelerate secure software development, Aviator uses large language models (LLMs) to analyze, enrich, and streamline SAST results, reducing triage time, improving fix quality, and enabling organizations to scale application security coverage without adding headcount.

Aviator automatically evaluates static scan results before any human review, identifying false positives, clarifying true issues, and adding contextual information that helps developers and auditors act faster. Each issue is accompanied by a plain language explanation of the risk, as well as remediation guidance, often in the form of suggested, copy ready code.

Designed with flexibility in mind, Aviator can be integrated directly into DevSecOps pipelines for inline automation or centrally enabled via Software Security Center (SSC). When deployed centrally, findings are automatically enriched and available to developers through existing workflows, transparent to the end user, and easy to scale across large portfolios.

This approach ensures high adoption, minimal friction, and consistent results across development teams, regardless of their preferred tools or languages.

Results at scale

Aviator was first proven against OpenText's own environment; a large, heterogeneous codebase where any weakness in scale or accuracy would surface quickly. Over the first eight weeks of internal deployment, 1,500 applications were onboarded.

It delivered measurable results:

300,000 findings analyzed and dispositioned

Three million minutes of manual review removed from the pipeline

Mean time to triage (MTTT) reduced by 70%

Developer satisfaction scores for security reviews improved significantly in internal surveys

"We built Aviator for customers like us. Large, complex engineering organizations that can't afford bottlenecks. Running it internally keeps us sharp."

Dylan Thomas,
Senior Director, Software Engineering, OpenText

With an average review time of 10 minutes per issue, **we saved more than three million minutes, equivalent to 50,000 hours or 2,080 days**. This translated to a productivity gain of **29 full-time employees across OpenText's engineering organization**, enabling teams to focus on delivering features instead of manual triage.

These gains avoided costs across multiple categories, including manual security audit, developer rework, delayed release timelines, and the productivity drain of constant context switching.

What drove these results was not just automation, but a developer centric experience. Aviator presents clear issue explanations and actionable, copy-paste remediation guidance directly within the tools developers already use. This reduced cognitive load and review fatigue, helping teams stay focused and engaged.

Tight integration with the existing developer toolchain accelerated adoption and reinforced consistency across teams. Familiar patterns, high trust in results, and low friction in daily workflows helped Aviator scale organically across the engineering organization.

> **"We're not just scaling AppSec; we're scaling trust. Trust in our process, in our AI, and in our ability to securely deliver innovation to market faster."**
>
> Dylan Thomas,
> Senior Director, Software Engineering, OpenText

Broader context: OpenText, AI, and the future of AppSec

OpenText Application Security is a recognized leader in Application Security Testing (AST), delivering a comprehensive platform that includes static (SAST), dynamic (DAST), software composition (SCA), and API security testing. With support for **more than 33 programming languages and more than one million APIs**, the OpenText Application Security platform provides deep, scalable coverage across modern development environments.

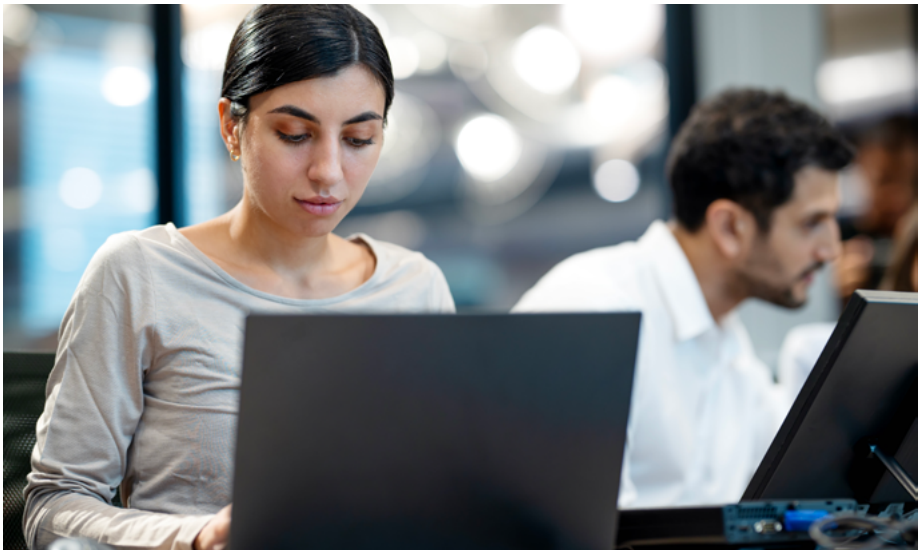
Aviator builds on this foundation with responsible AI that tackles one of the most persistent challenges in application security—**how to reduce false positives at scale without sacrificing accuracy**. By delivering explainable outputs in plain language and suggesting actionable fixes, Aviator enhances developer confidence in security findings and accelerates secure development practices across the software development lifecycle.

As AI reshapes the software landscape, OpenText is committed to using it to improve, not replace, human judgment. Aviator exemplifies this approach by acting as a force multiplier for application security teams, reducing triage fatigue, streamlining workflows, and enabling consistent enforcement of security standards across large portfolios.

What's next: Autonomous remediation and beyond

With triage acceleration now operational at scale, OpenText is entering the next phase of AI powered application security: **autonomous remediation**.

Building on the success of Application Security Aviator, we are preparing to roll out automated code fixes for suitable low and medium severity findings, starting with the most common patterns where safe, deterministic remediations can be applied. The goal is to further reduce the time and effort required to close out security issues, especially those that often linger in backlogs and erode risk posture over time.



Ready to see what AI-powered security looks like in practice?

Request a live demo to discover how Aviator can help reduce triage time, accelerate remediation, and scale secure development across your organization.

This is more than a tooling upgrade, it is a process evolution. By embedding automated remediation into the same workflows where Aviator currently enriches findings, we enable a seamless transition from identification to resolution. Teams will have the option to review, accept, or override suggested code changes, ensuring developer control and auditability remain intact.

In parallel, OpenText is enhancing Aviator's capabilities with:

- Improved remediation coverage across languages and vulnerability types.
- Automated enablement for new applications via Software Security Center.
- Ongoing tuning of prompt design and policy thresholds to ensure high confidence and explainability.
- Feedback loops from developers and security champions to refine outputs and expand trust in the system.

These updates reinforce our belief that scalable application security is not just about shifting left, it is about *shifting smart*. By combining deep static analysis with AI driven triage and remediation, OpenText is building the blueprint for intelligent, sustainable security at enterprise scale.

Conclusion

The rollout of Application Security Aviator at OpenText shows what is possible when AI, automation, and process come together in service of secure, scalable software development. It represents a new generation of AppSec strategy—one that embeds security throughout the development lifecycle, improves developer experience, and transforms compliance into a competitive advantage.

With Aviator, OpenText is not just imagining the future of application security. We are operating it today.

opentext™