BLACKDUCK®

How Black Duck Drives Development Productivity, Lowers Risk, and Decreases Cost for Customers

Research conducted and verified by

UserEvidence

Software developers and leadership teams don't need to be sold on the importance of application security (AppSec). But the development slowdown created by manual or poorly integrated AppSec practices has driven many organizations to hold these processes until the end of the software development life cycle (SDLC), ultimately creating long-term inefficiency, resource waste, and risk exposure.

Development teams know there must be a better way, the challenge is figuring out how to implement AppSec in a manner that optimizes both effort and investment.

This is what "shifting AppSec everywhere" is all about: embedding testing and security processes earlier and throughout the SDLC, with integrated and automated testing tools so that teams aren't caught off-guard at the finish line. It's also exactly what Black Duck helps customers achieve.

"It's all about starting the screening for security defects as early as possible," says Patrick Carey, Executive Director of Product Marketing at Black Duck. ***"It not only reduces downstream defects, it's just a more efficient way of working."***

To understand how organizations are bridging the divide between their development and security teams when implementing AppSec in their SDLCs, UserEvidence surveyed more than 100 Black Duck customers from a range of organizations around the globe, including software companies, hardware manufacturers, and government agencies.

This report uncovers the value that Black Duck solutions generate when securing applications, managing software supply chains, protecting sensitive data, and safeguarding valuable intellectual property.

Methodology

Black Duck commissioned independent market research agency, UserEvidence, to survey development leaders and stakeholders at organizations with development teams averaging 111 members.

Titles

VP of Technology | Lead Developer | Security Architect | Quality Analyst | Chief Security Information Officer | Product Manager | DevOps Engineer

Development team size

25%
<25 developers

12.5%
25-75 developers

16.3%
75-125 developers

9.6%
125-200 developers

5.8%
200-350 developers

30.8%
350+ developers

The survey focused on three core areas:

Productivity

How Black Duck impacts how much time developers spend on manual tasks and triage versus writing new code

Risk

How earlier and more frequent testing enables more complete security coverage and decreases overall software risk

Cost

How Black Duck decreases cost for customers by ensuring software releases are on time, secure, and high quality

In the end, UserEvidence collected feedback from more than 100 respondents across a wide range of roles spanning application development, testing, and security, with titles ranging from team leads to C-suite members. Here's what these respondents reported about the current and future impact of implementing Black Duck for their AppSec needs.

How Black Duck Improves Productivity

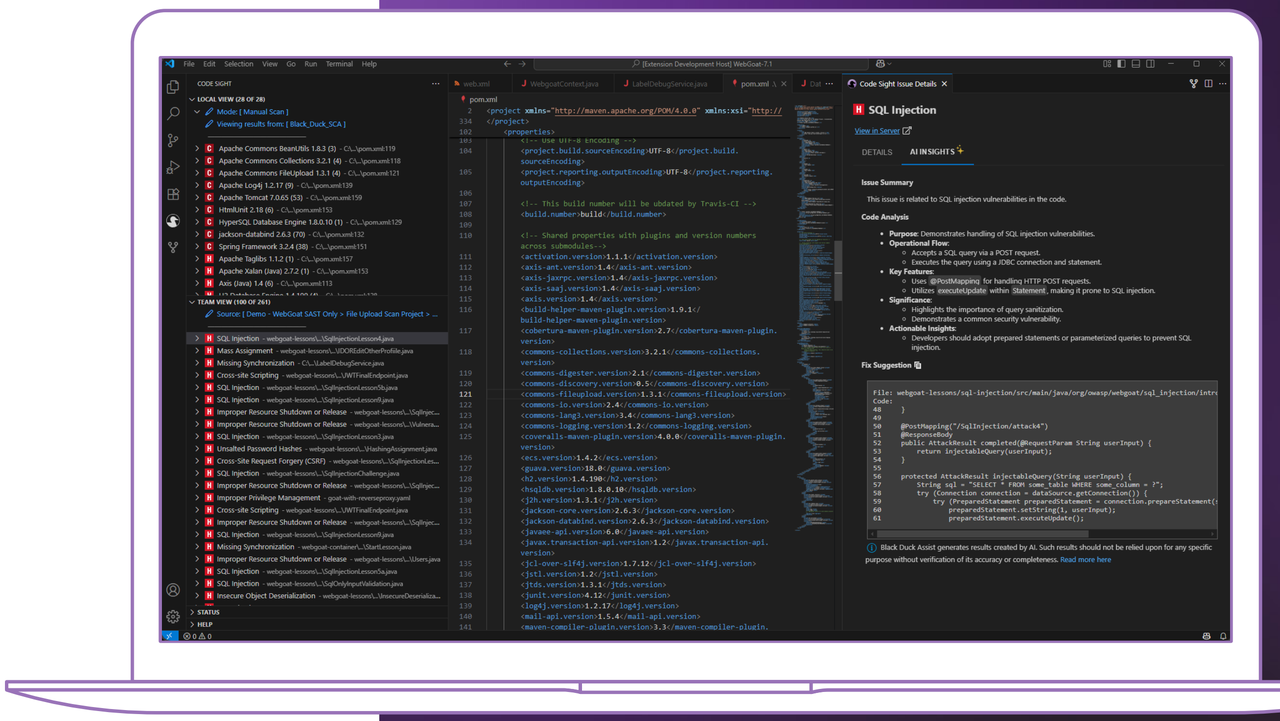
The Challenge: Supporting Developers to Work Efficiently at Scale

Developers already balance a complex tech stack, with each tool bringing its own learning curve, login credentials, and data silo. It's no surprise that introducing a security tool, especially if it goes against the grain of how developers usually work, can face pushback.

According to Ryan English, North America Director of Cybersecurity Sales at Black Duck, developers want something that plugs into their current ecosystem. "***The last thing they want is to go to a different system for security,***" he explains. ***"That's why Black Duck integrates with integrated development environments [IDEs], code repos, build servers, and defect repositories."***

The complexity of these tech stacks also complicates enterprise efforts to establish development timelines and guarantee productivity outcomes, especially at scale. But the difficulty of scaling AppSec isn't usually because development teams can't make it happen—instead, inconsistent or manual processes are often to blame.

This is a prominent issue among enterprises whose diverse business units, global nuances, and elaborate M&A landscapes establish the foundation upon which AppSec and development leads must construct a development, security, and operations (DevSecOps) program. When AppSec is implemented at the end of an established development process or bolted on to an existing workflow without being purposely integrated, it introduces friction. And anything that adds friction to a development team that is already being pushed to release code faster than ever before is just not sustainable.



Black Duck Reduces Manual Work

The result of security processes that are bolted on or managed outside of a typical development workflow is manual work for development and security teams. When teams use Black Duck to integrate and automate testing within the normal flow of development, that automation removes a heavy burden from the people involved.

Thus, when developers can rely on automated scans and integrated results, they spend less time having to manually kick off security processes or triage issues coming from outside their typical development tools. Case in point:

The average time Black Duck users spend weekly on manual code quality and security reviews typically drops by more than 42% after implementation.

The automation capabilities built into Black Duck solutions and the customizable workflow templates for software code management (SCM) and continuous integration (CI) platforms (e.g., GitHub Actions, GitLab Templates, Bitbucket Pipes) can also help teams act quickly and decrease the amount of manual work required. As Jeff Delaney, VP of Engineering at Black Duck, explains, ***"If developers find a vulnerability in an open source component—for example, Log4j 2.0—Black Duck flags it but also offers upgrade guidance, such as recommending version 2.2 if it's free of critical or high-severity issues."***

This dramatically decreases the amount of manual investigation required by the developer and greatly improves their overall productivity.

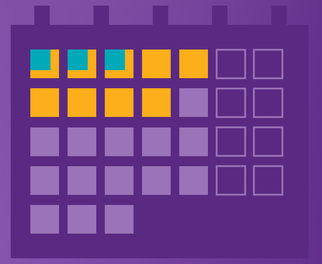
Black Duck Improves Remediation Times

There are two key aspects to consider when trying to remediate issues efficiently and effectively. First, at what point is the developer being alerted to an issue? If it's as they're coding, it's much faster for them to remediate in the moment than if they're alerted to an issue after they've moved on. However, this decision comes at a cost.

"There's developer velocity—how fast developers can write code—and delivery velocity—how fast that code actually reaches customers," explains Carey. ***"Teams focusing only on how quickly they get features live might feel like they're moving quickly, but if they uncover a pile of defects late in the cycle, it can stall a delivery. That's why layering security throughout the development process is key."***

Second, do developers have the information they need to quickly fix the issue, or do they have to spend time investigating a fix? Developers are not—nor should we expect them to be—security experts. To bridge this gap, Black Duck solutions provide them with remediation guidance and precise issue locations (e.g., affected project files, contributing lines of code), abbreviating risk investigation and triage, accelerating fixes, and reducing the opportunity for an attack or missed deadline.

This efficiency isn't just anecdotal: Black Duck customers see their remediation time drop nearly 66% after implementing Black Duck solutions.



66%

Drop in average remediation time after implementing Black Duck solutions.

Black Duck Increases Time Spent on Revenue-Generating Work

Developers' primary focus is to write new code, but they don't have time to do that with issue triage, late-stage fixes, and costly security audits competing for their time. Black Duck gives them that time back.

By using Black Duck to eliminate manual tasks, automate time-consuming rework, suggest quick fixes, and reduce the need to backtrack for things like patch releases, developers gain an extra 4.23 hours a week that they can spend writing new code.

That's 22% more time coding instead of chasing down false positives, rewriting code, or investigating issues in code they've moved on from.

"Reducing the number of patch releases and other fixes resulting from late-stage discoveries has a huge impact on the development life cycle," shares Delaney. ***"It lets developers spend more time writing new features and code instead of abruptly stopping to produce a patch release."***

The biggest productivity gains come from alerting developers to code or security issues the moment they happen and in the tools they're already using, not weeks or months later when they're getting ready to ship a product or are deep into the next release.

When vulnerabilities are caught during development, developers can address them while the context remains fresh. As Michael Knight, VP of Technology at DataScan and a Black Duck customer, puts it, ***"Getting the scan results faster means we can get them to development before the team switches their headspace. It avoids context switching and makes the process smoother for everyone."***

Carey agrees. ***"Context switching is a major pain for developers. If a developer finds a bug weeks later, they have to pause what they're doing, reset their environment, and get their head back into it."***

+4 hours
Average time developers get back per week to spend writing new code

19 hours
Average time developers spent per week on new code production before implementing Black Duck

23 hours
Average time developers spent per week on new code production after implementing Black Duck

How Black Duck Decreases Software Risk

The Challenge: Reducing Risk Exposure Through Early Detection

The business impact of poor, inefficient, or circumvented AppSec processes extends far beyond a drag on productivity. By placing the burden of AppSec on developers, organizations suffer from increased frustration and burnout that continues to compound as teams grow and release cycles shorten. An underestimated consequence of these internal issues is the formation of unapproved, unsecured secondary development pipelines outside the visibility of security teams that make adjusting AppSec tooling and policies difficult and proper risk assessment impossible.

Again, developers are not security experts, and when they're relied upon to perform security functions, organizations undercut their own risk mitigation efforts and expose themselves to future vulnerabilities that come at a significant productivity and revenue cost.

There are a few challenges that teams must contend with to reduce their risk exposure.

Maximizing security test coverage

Run the right tests at the right time for maximum coverage and minimal disruption

Minimizing defect rates

Leverage test coverage throughout the SDLC to make sure critical issues do not make their way to production

Ensuring complete visibility

Pinpoint the latest headline-catching issue, attest to its resolution, and efficiently report on software risk

Our team saw a clear improvement in code quality with Black Duck. It identified critical and high-severity issues, including ones our previous scanner missed, so we know we're in a better place."

– Michael Knight, VP of Technology at DataScan and a Black Duck customer

With Black Duck, we provide a precise bill of materials that details what open source you're using, which versions, and the associated vulnerabilities. That level of visibility is critical for compliance and security."

– Jeff Delaney, VP of Engineering at Black Duck

Black Duck Increases Security Coverage

One of the most critical AppSec requirements is knowing that you have your bases covered and the mechanisms in place to act as a safety net when a human is not involved. This means being able to affirmatively answer questions like the following:

- Do you have complete visibility into every open source component and its associated risks?
- Are you catching vulnerabilities across your software supply chain and into runtime?
- Are you able to catch new risks as they arise?
- Is your code high quality, meaning your applications maintain performance and uptime?

When AppSec is seen as an inhibitor rather than an enabler, teams are forced to pick and choose where they run tests. But when you can implement automated scans and deliver high-quality results in context, coverage can naturally be extended.

Knight shares that his team saw ***"a clear improvement in code quality with Black Duck. It identified critical and high-severity issues, including ones our previous scanner missed, so we know we're in a better place."***

This is the kind of coverage Black Duck delivers for customers. By running more thorough scans across all major AppSec categories and continuously throughout the SDLC, customers not only find and fix security, quality, and IP issues quickly, they can also generate a comprehensive software bill of materials (SBOM) for complete and continued software visibility.

Survey data backs this up: Black Duck customers report a 40% average increase in security coverage since implementing Black Duck solutions.

Expanded coverage across the SDLC not only multiplies the impact of AppSec automation, it closes the loop with developers at natural points in their existing workflows to facilitate faster remediation before software is promoted into production.

Black Duck Minimizes Security Defects in Production

Atop the list of “worst-case scenarios” according to AppSec teams is a high-severity vulnerability that leads to a major security breach.

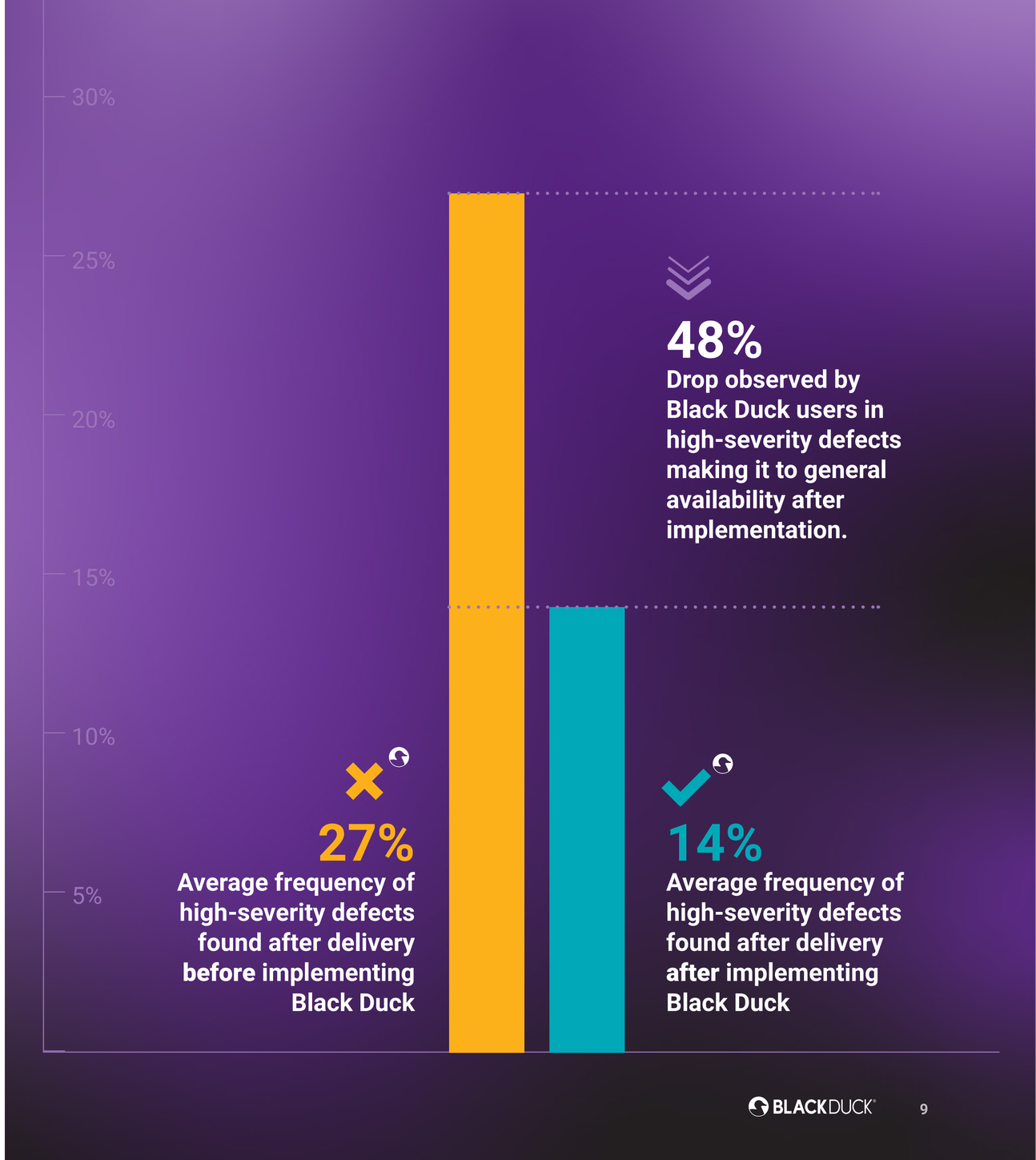
Whether or not this high-stakes issue leads to a breach, an emergency patch, a hotfix, or a code rollback, it will derail the current release (and the next one), creating major downstream implications for internal contributors, external stakeholders, and ultimately customers.

Unfortunately, this is all too common for AppSec teams, with survey respondents reporting that **before implementing Black Duck, teams encountered high-severity defects in production 27% of the time**. In other words, more than 1 in 4 releases still contained serious issues after being shipped to customers, and moreover, these issues were not detected early enough to be remedied prior to production.

This is where solutions like Black Duck make all the difference. By shifting testing earlier in the build process and detecting issues when developers still have the context and time to act, organizations can reduce the number of last-minute fire drills and prevent potential breaches and consequent fallout.

In fact, **Black Duck customers see a 48% drop in high-severity defects making it to production after implementation.**

Not only did fewer high-severity defects make their way to production, but customers also saw a nearly 24% drop in overall security defects on average.

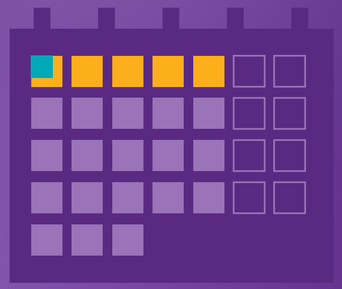


Black Duck Streamlines Risk Reporting

Some of the most time-consuming parts of AppSec include the risk reporting and security audits that need to be presented back to the business or to auditors. These tasks, however tedious, are requisite aspects of participating in expansive software supply chains where an organization must both verify the risk status of assets it ingests and of the software it ships to customers and downstream partners.

Before implementing Black Duck, the average time to prepare risk reports or perform security audits was 5.06 days. After implementing Black Duck? That number dropped to 1.24, representing a 75% decrease.

Multiply that time savings across multiple audits and the teams involved, and organizations win back dozens of hours that leaders can reinvest in higher-ROI activities.



75%

Average drop in time to prepare risk reports or perform security audits after implementing Black Duck tools

CASE STUDY

What Sets Black Duck's Risk Mitigation Capabilities Apart

When it came time for Knight and his team to select a solution to meet their application security needs, Black Duck was the obvious choice.

According to Knight, Black Duck delivers three critical capabilities for his team:

Flexible scanning options

Seamless integration into existing workflows

Fast, accurate results

With Black Duck solutions, you're not just getting AppSec tools that work, you're getting tools that work with you, and that's what sets Black Duck apart.

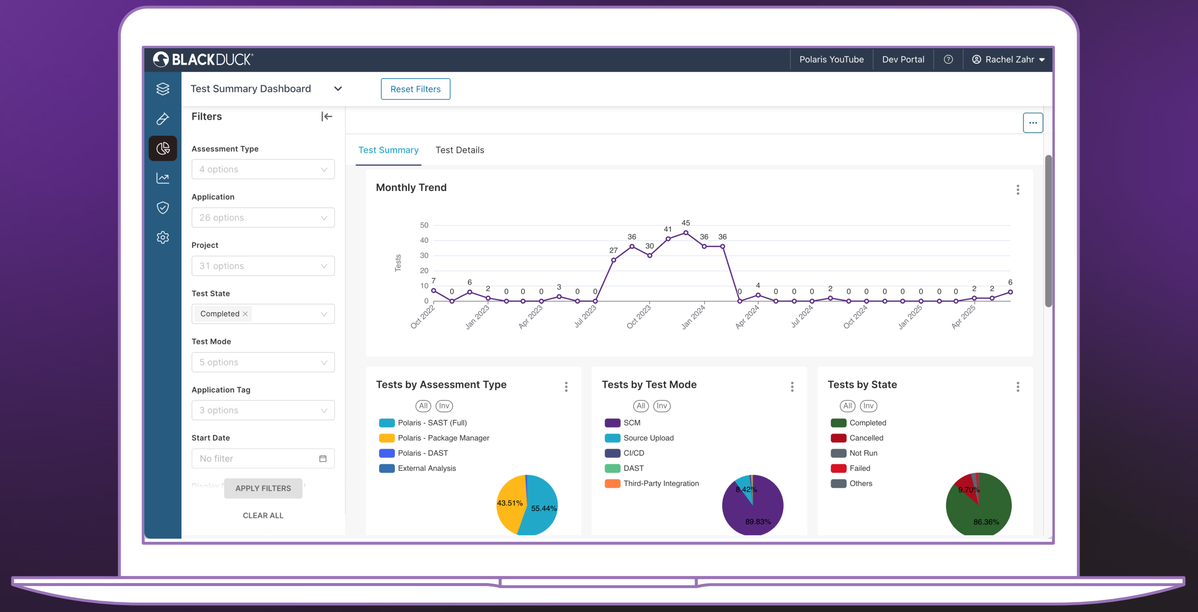
Flexible Scanning That Meets Teams Where They Are

Every team works differently, and no two release cycles are the same, which is why Knight points to Black Duck's flexible scanning options as a major win. ***"Black Duck's ability to align to security-defined policies while functioning on automated pipeline triggers—things like code commits, pull requests, and builds—means that scans can run as early as possible while accommodating project nuances, contextual changes, and risk tolerance."***

For Knight and his team, using the Polaris Platform has enabled them to scale automatically as they grow, with the SaaS deployment decreasing overhead and reducing overall cost. For growing teams like Knight's, this adaptability and flexibility have been key. ***"We're scanning a lot more now, especially as our teams and product lines grow,"*** he says. ***"Running multiple scans in parallel helps us keep up during the QA process and release crunches."***

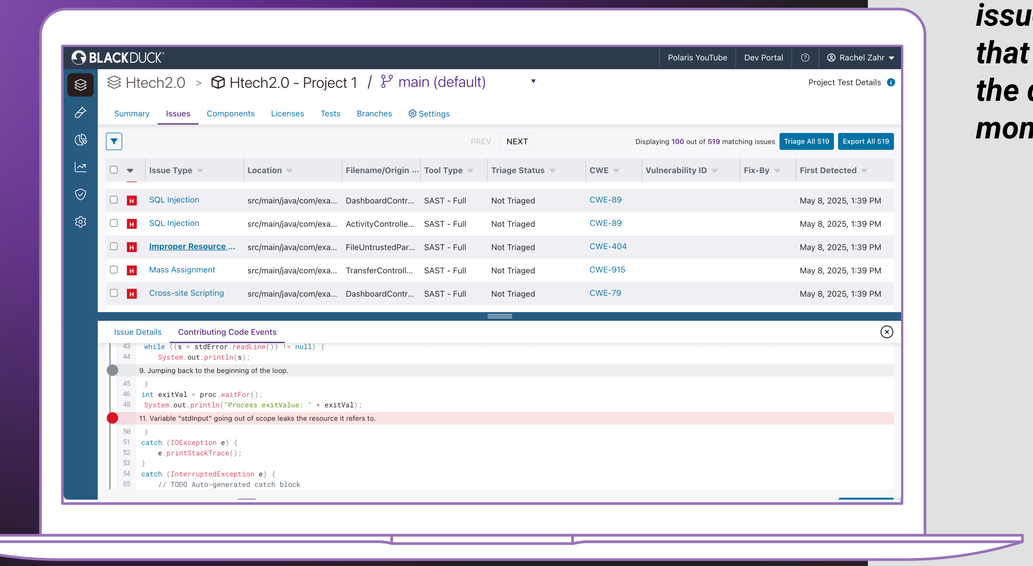
This kind of flexibility supports a broader shift toward continuous security testing, which Patrick Carey, Executive Director of Product Marketing at Black Duck, says is key to short- and long-term success. ***"That's where this idea of amortizing the investment in AppSec—starting early but testing continuously—comes into play,"*** he explains. ***"You don't create bottlenecks, you manage the flow throughout."***

> **"We're scanning a lot more now, especially as our teams and product lines grow. Running multiple scans in parallel helps us keep up during the QA process and release crunches."**
>
> *– Michael Knight, VP of Technology at DataScan and a Black Duck customer*



Being able to report issues while developers are still working on that particular issue is huge. It's the difference between taking care of it at the point the developer is working on it or circling back months later."

– Michael Knight, VP of Technology at DataScan and a Black Duck customer



Building Risk Tolerance Into Existing Developer Workflows

One of the hardest parts of rolling out any new tool to internal teams—AppSec-related or otherwise—is getting teams to use it. Major barriers to adoption arise when the tool is inconvenient, clunky, or disconnected from existing workflows.

For Knight and his team, Black Duck fit into their workflow from the start. ***"Being able to report issues while developers are still working on that particular issue is huge,"*** he says. ***"It's the difference between taking care of it in the moment or circling back months later."***

Producing Fast, Accurate Results

Legacy methods of performing batched scans overnight or at the end of the week lengthen scan times and delay triage and remediation. Additionally, inefficient scan engines or poorly configured integrations can cause unnecessary friction or break build pipelines, all of which contributes to a painful degradation of operational efficiency, which was a significant pain point for Knight's team with a previous vendor.

As Knight explains, ***"With Black Duck, the QA process is built into the scan windows, so there's no extra waiting required, and our development teams can start remediation efforts right when the scans are complete."***

Beyond optimized workflows and flexible scan configurations, Black Duck provides extensive, powerful APIs and command-line interface capabilities to facilitate further automation and refine the management and invocation of testing engines.

How Black Duck Avoids the Cost of Delayed Releases

The Challenge: Late Detection Creates Extra Work at the Worst Time

After months, quarters, or even years of building, testing, and planning, a security risk or software license incompatibility that shows up at the last minute is a development team's worst nightmare. Why? Because it grinds everything to a halt. Instead of staying on schedule or getting the green light to start the next sprint, developers must flip the switch and fix what someone should have caught ages ago.

And these late-stage fixes aren't usually quick turnarounds either—not only must developers return to code they shipped long ago, they also have to reconfigure their environments, coordinate with other teams, document everything for audits, and ultimately rush a process that should be thoughtful and deliberate. In the case of an open source software license incompatibility, a developer must find a functionally analogous component to replace the piece that caused the license conflict, if one exists at all.

Often, despite the AppSec team's best attempts to refine results, testing tools can generate false positives or flag issues that are not exploitable at runtime. It's the software world's version of chasing ghosts.

And all of that chasing adds up, explains Ryan English, North America Director of Cybersecurity Sales at Black Duck. ***"False positives take time because developers must investigate and document them for auditors. All that work makes false positives surprisingly costly,"*** he shares.

"If Black Duck flags a SQL injection during development, the developer can fix it in minutes. But if it's caught three months later, it's unclear who wrote it, and someone has to investigate, regain context, fix the issue, and run additional tests. That all adds up and creates costly delays."

– Jeff Delaney, VP of Engineering at Black Duck

Average cost of a delayed release for a major project

62%
$50-100k

16%
$100-250k

12%
$250-500k

10%
$500k+

Black Duck Reduces the Likelihood of Delays, Saving Teams Money

Black Duck solutions empower security teams to prioritize true risks and apply risk tolerance policies that minimize wasted effort on inconsequential issues. An AppSec team may be willing to accept a few medium-severity vulnerabilities within applications that run behind a firewall, for example, or they may deprioritize issues in source files used only in development and not invoked in the final build.

Focusing on true positives and optimizing risk triage will not only have a positive impact on internal teams' productivity, it can also have a tangible impact on the bottom line by helping organizations achieve the following:

- Deliver competitive and differentiating software enhancements more quickly and consistently
- Avoid costly application downtime and maximize uptime to drive customer satisfaction and solution reliability
- Shift AppSec from a cost center to a revenue engine, establishing a reputation within the software supply chain as a security-capable vendor and among customers as a safe harbor for sensitive data

In our survey, **nearly 4 in 10 respondents said that the average cost of a delayed release to a major project surpassed $100,000. Of those, almost a quarter reported losses of more than $500,000.**

In fact, a large software company estimates that before implementing Black Duck, >75% of releases were delayed due to security. Each delayed release cost the company $500k+.

Since implementing Black Duck, only 10-25% of releases are delayed, saving the company millions of dollars a year.

This is where the true cost of pushing AppSec to the end of the development life cycle, to avoid impeding development velocity, adds up.

And these self-reported numbers don't necessarily capture the full cost of delayed major projects.

"A delayed release is extremely costly," explains Delaney. ***"It's hard to put a number on it because it depends on how big the release is. But when you have 100 developers on a product, and you delay by two weeks, it affects that release and has a knock-on effect on the next release."***

To put this in perspective, if the average fully loaded developer costs a company between $120,000 and $150,000 a year, the company's paying about $2,600 per week per developer. Multiply that by 100 developers, and a **two-week delay could burn through $520,000 in payroll alone,** more than half a million dollars in lost productivity and value the company won't get back.

Factor in disrupted release schedules, unfulfilled promises to customers, blocked dependencies for other teams, the increased burden on QA and customer support, and that dollar amount can easily multiply. There's also the missed revenue due to delayed launches, missed upsells, or lost trust that could send a customer to a competitor.

"Technically, nothing's ever late," says English. ***"Every company always hits its release dates, but features fall out of scope, and you can lose customers over this."***

Knight offers a clearer example. ***"If we delay a release by even a month, the financial impact is significant,"*** he explains. ***"Our ability to generate revenue depends on customers using the features we deliver. If those features are delayed or incomplete, they can't fully operate on our platform—and that directly affects our bottom line."***

Thanks to Black Duck's accurate scan results, customizable policy controls, and detailed fix guidance, **customers report a 55% reduction in delayed releases due to security issues.**

55%

Average drop in delayed releases due to security issues after implementing Black Duck tools

If we delay a release by even a month, the financial impact is significant. Our ability to generate revenue depends on customers using the features we deliver. If those features are delayed or incomplete, they can't fully operate on our platform—and that directly affects our bottom line.

– Michael Knight, VP of Technology at DataScan and a Black Duck customer

Development Speed and Security Aren't Mutually Exclusive

Security, velocity, and delivery don't have to compete. The highest-performing teams check all three boxes—but not by cutting corners.

By helping organizations efficiently integrate AppSec processes throughout their development workflows, Black Duck AppSec solutions support optimized development life cycles that

- Accelerate your time to market
- Avoid costly release delays
- Increase developer capacity for revenue-generating projects
- Reduce risk exposure
- Improve your overall risk posture

Remember: 95% of Black Duck customers report a decrease in overall software risk since implementing Black Duck tools.

Your organization could be next.

Contact us today to get started.

www.blackduck.com

About UserEvidence

UserEvidence is a software company and independent research partner that helps B2B technology companies produce original research content from practitioners in their industry. All research completed by UserEvidence is verified and authentic according to their research principles: Identity verification, significance and representation, quality and independence, and transparency. All UserEvidence research is based on real user feedback without interference, bias, or spin from our clients.

UserEvidence Research Principles

These principles guide all research efforts at UserEvidence—whether working with a vendor's users for our Customer Evidence offering, or industry practitioners in a specific field for our Research Content offering. The goal of these principles is to give buyers trust and confidence that you are viewing authentic and verified research based on real user feedback, without interference, bias, and spin from the vendor.

Principle 1 — Identity verification.

In every study we conduct, UserEvidence independently verifies that a participant in our research study is a real user of a vendor (in the case of Customer Evidence) or an industry practitioner (in the case of Research Content). We use a variety of human and algorithmic verification mechanisms, including corporate email domain verification (i.e., so a vendor can't just create 17 Gmail addresses that all give positive reviews).

Principle 2 — Significance and Representation.

UserEvidence believes trust is built by showing an honest and complete representation of the success (or lack thereof) of users. We pursue statistical significance in our research and substantiate our findings with a large and representative set of user responses to create more confidence in our analysis. We aim to canvas a diverse swatch of users across industries, seniorities, personas—to provide the whole picture of usage and allow buyers to find relevant data from other users in their segment, not just a handful of vendor-curated happy customers.

Principle 3 — Quality and Independence.

UserEvidence is committed to producing quality and independent research at all times. This starts at the beginning of the research process with survey and questionnaire design to drive accurate and substantive responses. We aim to reduce bias in our study design and use large sample sizes of respondents where possible. While UserEvidence is compensated by the vendor for conducting the research, trust is our business and our priority, and we do not allow vendors to change, influence, or misrepresent the results (even if they are unfavorable) at any time.

Principle 4 — Transparency.

We believe research should not be done in a black box. For transparency, all UserEvidence research includes the statistical N (number of respondents), and buyers can explore the underlying blinded (de-identified) raw data and responses associated with any statistic, chart, or study. UserEvidence provides clear citation guidelines for clients when leveraging research, including guidelines on sharing research methodology and sample size.

About Black Duck

Black Duck meets the board-level risks of modern software with True Scale Application Security, ensuring uncompromised trust in software for the regulated, AI-powered world. Only Black Duck solutions free organizations from tradeoffs between speed, accuracy, and compliance at scale while eliminating security, regulatory, and licensing risks. Whether in the cloud or on premises, Black Duck is the only choice for securing mission-critical software everywhere code happens. With Black Duck, security leaders can make smarter decisions and unleash business innovation with confidence. Learn more at www.blackduck.com.